



**Concepts & Examples
ScreenOS Reference Guide**

**Volume 9:
User Authentication**

Release 6.1.0, Rev. 03

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Copyright Notice

Copyright © 2009 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, ScreenOS, and Steel-Belted Radius are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOSe is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

All specifications are subject to change without notice. Juniper Networks assumes no responsibility for any inaccuracies in this document or for any obligation to update information in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

FCC Statement

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. The equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Juniper Networks' installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Consult the dealer or an experienced radio/TV technician for help.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

Caution: Changes or modifications to this product could void the user's warranty and authority to operate this device.

Disclaimer

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR JUNIPER NETWORKS REPRESENTATIVE FOR A COPY.

Table of Contents

	About This Volume	vii
	Document Conventions	viii
	Web User Interface Conventions	viii
	Command Line Interface Conventions	viii
	Naming Conventions and Character Types	ix
	Illustration Conventions	x
	Requesting Technical Support	x
	Self-Help Online Tools and Resources	xi
	Opening a Case with JTAC	xi
	Document Feedback	xi
Chapter 1	Authentication	1
	User Authentication Types	1
	Admin Users	2
	Multiple-Type Users	4
	Group Expressions	5
	Example: Group Expressions (AND)	6
	Example: Group Expressions (OR)	8
	Example: Group Expressions (NOT)	9
	Banner Customization	10
	Example: Customizing a WebAuth Banner	10
	Login Banner	10
	Example: Creating a Login Banner	11
Chapter 2	Authentication Servers	13
	Authentication Server Types	13
	Local Database	15
	Example: Local Database Timeout	16
	External Authentication Servers	17
	Auth Server Object Properties	18
	Auth Server Types	19
	Remote Authentication Dial-In User Service	19
	RADIUS Auth Server Object Properties	20
	Supported User Types and Features	20
	RADIUS Dictionary File	21
	Supported RADIUS Enhancements for Auth and XAuth Users	24
	SecurID	27
	SecurID Auth Server Object Properties	28
	Supported User Types and Features	28
	Lightweight Directory Access Protocol	29
	LDAP Auth Server Object Properties	30
	Supported User Types and Features	30
	Terminal Access Control Access Control System Plus (TACACS+)	30

	TACACS+ Server Object Properties	32
	Prioritizing Admin Authentication	32
	Defining Auth Server Objects	33
	Example: RADIUS Auth Server	33
	Example: SecurID Auth Server	35
	Example: LDAP Auth Server	36
	Example: TACACS+ Auth Server.....	38
	Defining Default Auth Servers	39
	Example: Changing Default Auth Servers	39
Chapter 3	Infranet Authentication	41
	Unified Access Control Solution	42
	How the Security Device Works with the Infranet Controller	43
	Dynamic Discovery of Infranet Enforcers	44
	Infranet Controller Clustering.....	45
Chapter 4	Authentication Users	47
	Referencing Auth Users in Policies	48
	Run-Time Authentication.....	48
	Pre-Policy Check Authentication (WebAuth)	49
	Referencing Auth User Groups in Policies	50
	Example: Run-Time Authentication (Local User)	51
	Example: Run-Time Authentication (Local User Group)	52
	Example: Run-Time Authentication (External User)	54
	Example: Run-Time Authentication (External User Group)	55
	Example: Local Auth User in Multiple Groups	58
	Example: WebAuth (Local User Group).....	60
	Example: WebAuth (External User Group)	61
	Example: WebAuth + SSL Only (External User Group)	63
Chapter 5	IKE, XAuth, and L2TP Users	67
	IKE Users and User Groups	67
	Example: Defining IKE Users.....	68
	Example: Creating an IKE User Group	69
	Referencing IKE Users in Gateways	70
	XAuth Users and User Groups	70
	Event Logging for IKE Mode	71
	XAuth Users in IKE Negotiations.....	72
	Example: XAuth Authentication (Local User)	73
	Example: XAuth Authentication (Local User Group)	75
	Example: XAuth Authentication (External User)	76
	Example: XAuth Authentication (External User Group).....	78
	Example: XAuth Authentication and Address Assignments (Local User Group)	81
	XAuth Client	85
	Example: Security Device as an XAuth Client.....	85
	L2TP Users and User Groups.....	86
	Example: Local and External L2TP Auth Servers.....	86
Chapter 6	Extensible Authentication for Wireless and Ethernet Interfaces	91
	Overview	92
	Supported EAP Types.....	92

Enabling and Disabling 802.1X Authentication	93
Ethernet Interfaces	93
Wireless Interfaces	93
Configuring 802.1X Settings.....	94
Configuring 802.1X Port Control	94
Configuring 802.1X Control Mode	95
Setting the Maximum Number of Simultaneous Users.....	95
Configuring the Reauthentication Period	96
Enabling EAP Retransmissions	96
Configuring EAP Retransmission Count.....	97
Configuring EAP Retransmission Period	97
Configuring the Silent (Quiet) Period	97
Configuring Authentication Server Options	98
Specifying an Authentication Server	98
Ethernet Interfaces.....	98
Wireless Interfaces.....	98
Setting the Account Type.....	99
Enabling Zone Verification.....	99
Viewing 802.1X Information.....	100
Viewing 802.1X Global Configuration Information	100
Viewing 802.1X Information for an Interface	100
Viewing 802.1X Statistics	100
Viewing 802.1X Session Statistics.....	101
Viewing 802.1X Session Details.....	102
Configuration Examples.....	102
Configuring the Security Device with a Directly Connected Client and RADIUS Server	102
Configuring a Security Device with a Hub Between a Client and the Security Device.....	103
Configuring the Authentication Server with a Wireless Interface	104
Index.....	IX-I

About This Volume

Volume 9: User Authentication describes the methods in ScreenOS for authenticating different types of users. It provides an introduction to user authentication, presents the two locations that can store user profiles—the internal database and an external authentication server—and provides numerous examples for configuring authentication, IKE, XAuth, and L2TP users and user groups. Some other aspects of user authentication are also covered, such as changing login banners, creating multiple-type users (such as an IKE/XAuth user, for example), and using group expressions in policies applying authentication.

This volume contains the following chapters:

- Chapter 1, “Authentication,” details the various authentication methods and uses that ScreenOS supports.
- Chapter 2, “Authentication Servers,” presents the options of using one of four possible types of external authentication server—RADIUS, SecurID, TACACS+ , or LDAP—or the internal database and shows how to configure the security device to work with each type.
- Chapter 3, “Infranet Authentication,” details how the security device is deployed in a unified access control (UAC) solution. Juniper Networks UAC secures and ensures the delivery of applications and services across an enterprise infranet.
- Chapter 4, “Authentication Users,” explains how to define profiles for authentication users and how to add them to user groups stored either locally or on an external RADIUS authentication server.
- Chapter 5, “IKE, XAuth, and L2TP Users,” explains how to define IKE, XAuth, and L2TP users. Although the XAuth section focuses primarily on using the security device as an XAuth server, it also includes a subsection on configuring select security devices to act as an XAuth client.
- Chapter 6, “Extensible Authentication for Wireless and Ethernet Interfaces,” explains the options available for and examples of how to use the Extensible Authentication Protocol to provide authentication for Ethernet and wireless interfaces.

Document Conventions

This document uses the conventions described in the following sections:

- “Web User Interface Conventions” on page viii
- “Command Line Interface Conventions” on page viii
- “Naming Conventions and Character Types” on page ix
- “Illustration Conventions” on page x

Web User Interface Conventions

The Web user interface (WebUI) contains a navigational path and configuration settings. To enter configuration settings, begin by clicking a menu item in the navigation tree on the left side of the screen. As you proceed, your navigation path appears at the top of the screen, with each page separated by angle brackets.

The following example shows the WebUI path and parameters for defining an address:

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: addr_1
 IP Address/Domain Name:
 IP/Netmask: (select), 10.2.2.5/32
 Zone: Untrust

To open Online Help for configuration settings, click the question mark (?) in the upper left of the screen.

The navigation tree also provides a Help > Config Guide configuration page to help you configure security policies and Internet Protocol Security (IPSec). Select an option from the list, and follow the instructions on the page. Click the ? character in the upper left for Online Help on the Config Guide.

Command Line Interface Conventions

The following conventions are used to present the syntax of command line interface (CLI) commands in text and examples.

In text, commands are in **boldface** type and variables are in *italic* type.

In examples:

- Variables are in *italic* type.
- Anything inside square brackets [] is optional.
- Anything inside braces { } is required.

- If there is more than one choice, each choice is separated by a pipe (|). For example, the following command means “set the management options for the ethernet1, the ethernet2, or the ethernet3 interface”:

```
set interface { ethernet1 | ethernet2 | ethernet3 } manage
```

NOTE: When entering a keyword, you only have to type enough letters to identify the word uniquely. Typing **set adm u whee j12fmt54** will enter the command **set admin user wheezer j12fmt54**. However, all the commands documented in this guide are presented in their entirety.

Naming Conventions and Character Types

ScreenOS employs the following conventions regarding the names of objects—such as addresses, admin users, auth servers, IKE gateways, virtual systems, VPN tunnels, and zones—defined in ScreenOS configurations:

- If a name string includes one or more spaces, the entire string must be enclosed within double quotes; for example:

set address trust “local LAN” 10.1.1.0/24
- Any leading spaces or trailing text within a set of double quotes are trimmed; for example, “ **local LAN** ” becomes “**local LAN**”.
- Multiple consecutive spaces are treated as a single space.
- Name strings are case-sensitive, although many CLI keywords are case-insensitive. For example, “**local LAN**” is different from “**local lan**”.

ScreenOS supports the following character types:

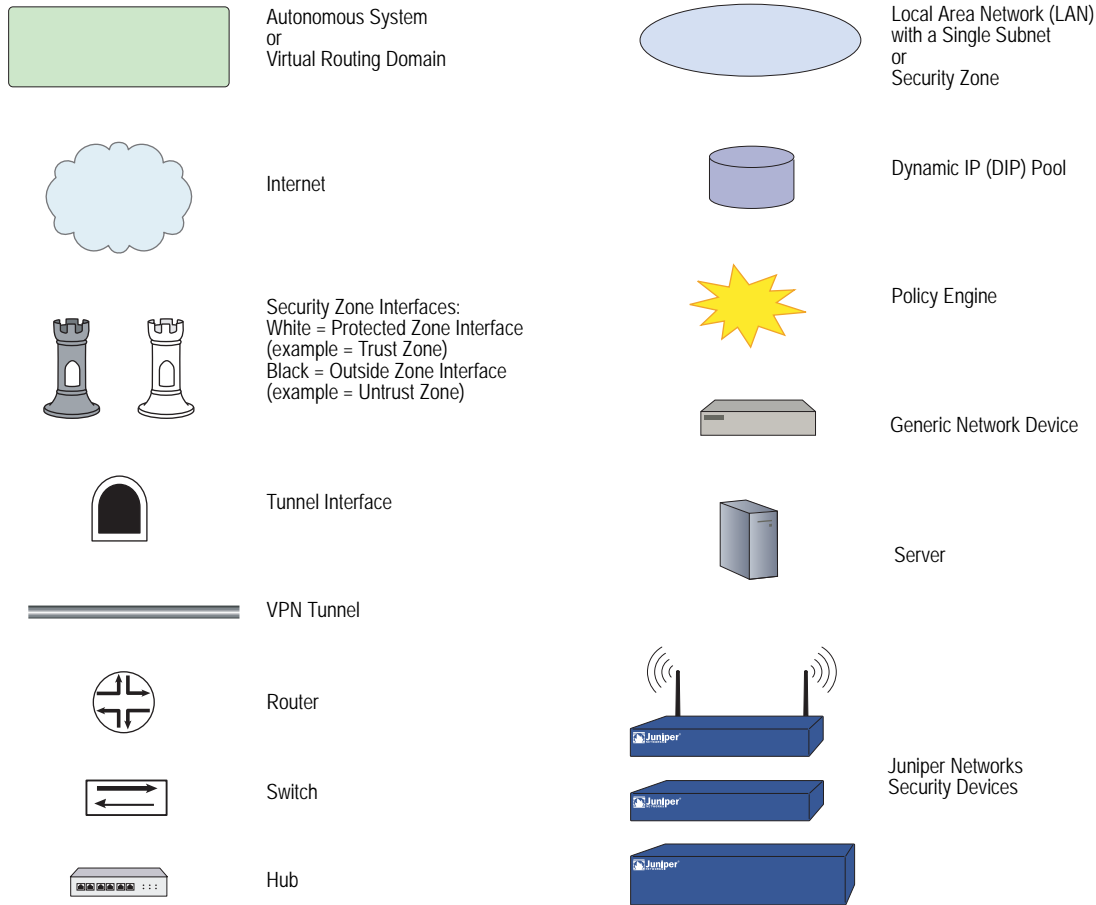
- Single-byte character sets (SBCS) and multiple-byte character sets (MBCS). Examples of SBCS are ASCII, European, and Hebrew. Examples of MBCS—also referred to as double-byte character sets (DBCS)—are Chinese, Korean, and Japanese.
- ASCII characters from 32 (0x20 in hexadecimal) to 255 (0xff), except double quotes (“), which have special significance as an indicator of the beginning or end of a name string that includes spaces.

NOTE: A console connection only supports SBCS. The WebUI supports both SBCS and MBCS, depending on the character sets that your browser supports.

Illustration Conventions

Figure 1 shows the basic set of images used in illustrations throughout this volume.

Figure 1: Images in Illustrations



Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/customers/support/downloads/710059.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings—<http://www.juniper.net/customers/support/>
- Find product documentation—<http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base—<http://kb.juniper.net/>
- Download the latest versions of software and review your release notes—<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications—<http://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum—<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Manager—<http://www.juniper.net/customers/cm/>
- To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool—<https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Manager tool in the CSC at <http://www.juniper.net/customers/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822—toll free in USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at <http://www.juniper.net/customers/support/requesting-support/>.

Document Feedback

If you find any errors or omissions in this document, contact Juniper Networks at techpubs-comments@juniper.net.

Chapter 1

Authentication

After a general introduction to the different types of authentication that are available for different types of network users, this chapter contains a brief section on admin user authentication. It then provides information on combining different user types, the use of group expressions, and how to customize the banners that appear on HTTP, FTP, L2TP, Telnet, and XAuth login prompts. The final section describes how to create a large, 4Kbyte banner that pre-empts all individually defined administrative access and firewall authentication banners. This chapter contains the following sections:

- “User Authentication Types” on page 1
- “Admin Users” on page 2
- “Multiple-Type Users” on page 4
- “Group Expressions” on page 5
- “Banner Customization” on page 10
- “Login Banner” on page 10

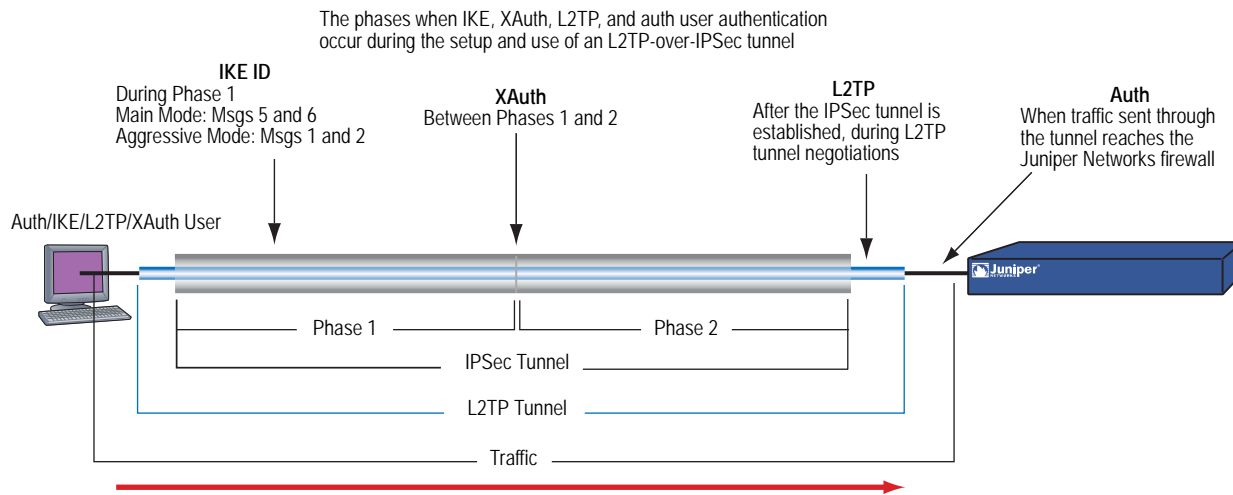
User Authentication Types

The following chapters describe the different types of users and user groups that you can create and how to use them when configuring policies, IKE gateways, and L2TP tunnels:

- “Authentication Users” on page 47
- “IKE Users and User Groups” on page 67
- “XAuth Users and User Groups” on page 70
- “L2TP Users and User Groups” on page 86

The security device authenticates the different types of users at different stages in the connection process. IKE, XAuth, L2TP, and auth user authentication techniques occur at different times during the creation of an L2TP-over-IPSec VPN tunnel. See Figure 2 on page 2.

Figure 2: Authentication During L2TP-over-IPSec VPN Tunnel



Note: Because XAuth and L2TP both provide user authentication and address assignments, they are seldom used together. They are shown together here solely to illustrate when each type of authentication occurs during the creation of a VPN tunnel.

Admin Users

Admin users are the administrators of a security device. There are five kinds of admin users:

- Root admin
- Root-level read/write admin
- Root-level read-only admin
- Vsys admin
- Vsys read-only admin

NOTE: For information about the privileges of each type of admin user and for examples of the creation, modification, and removal of admin users, see “Administration” on page 3-1.

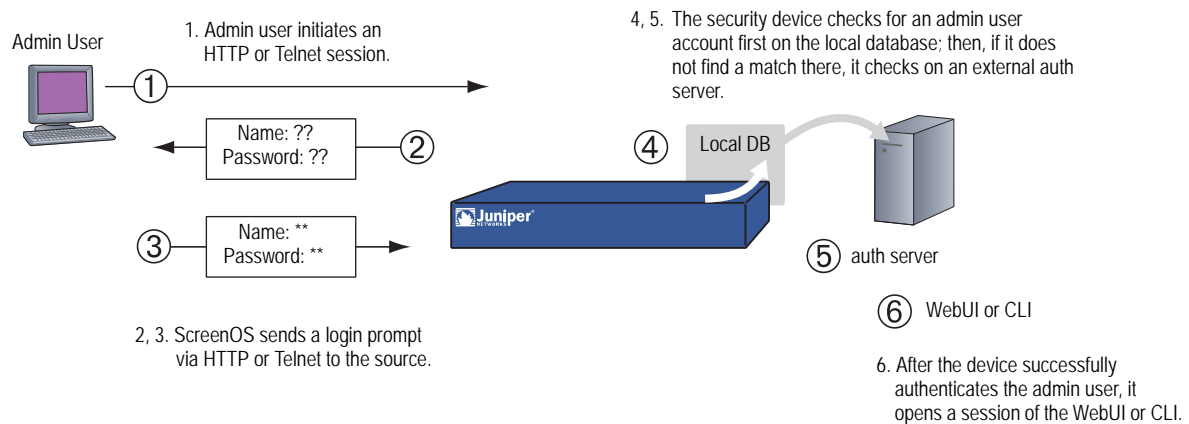
Although the profile of the root user of a security device must be stored in the local database, you can store vsys users and root-level admin users with read/write and read-only privileges either in the local database or on an external auth server.

If you store admin user accounts on an external RADIUS auth server and you load the RADIUS dictionary file on the auth server, you can elect to query admin privileges defined on the server. Optionally, you can specify a privilege level to be applied globally to all admin users stored on that auth server. You can specify either read/write or read-only privileges. If you store admin users on an external SecurID or LDAP auth server, or on a RADIUS server without the RADIUS dictionary file, you cannot define their privilege attributes on the auth server. Therefore, you must assign a privilege level to them on the security device.

If set on the security device:	And the RADIUS server is loaded with the RADIUS dictionary file, then:	And a SecurID, LDAP, or RADIUS server without the RADIUS dictionary file, then:
Get privileges from RADIUS server	Assign appropriate privileges	Root-level or vsys-level admin login fails
Assign read/write privileges to external admin	Assign root-level or vsys-level read/write privileges	Assign root-level read/write privileges Vsys admin login fails
Assign read-only privileges to external admin	Assign root-level or vsys-level read-only privileges	Assign root-level read-only privileges Vsys admin login fails

Figure 3 shows the admin authentication process.

Figure 3: Admin Authentication Process



Multiple-Type Users

You can combine auth, IKE, L2TP, XAuth users to create the following combinations to store on the local database:

- Auth/IKE user
- Auth/IKE/XAuth user
- Auth/L2TP user
- IKE/XAuth user
- Auth/IKE/L2TP user
- L2TP/XAuth user
- IKE/L2TP user
- IKE/L2TP/XAuth user
- Auth/XAuth user
- Auth/IKE/L2TP/XAuth user

Although you can make all of the above combinations when defining multiple-type user accounts on the local database, consider the following points before creating them:

- Combining an IKE user type with any other user type limits the potential to scale. You must store an IKE user account on the local database. If you create auth/IKE, IKE/L2TP, and IKE/XAuth user accounts and then the number of users grows beyond the capacity of the local database, you will not be able to relocate these accounts to an external auth server. If you separate IKE user accounts from other types of accounts, you have the flexibility to move the non-IKE user accounts to an external auth server should the need arise to do so.
- L2TP and XAuth provide the same services: remote user authentication and IP, DNS server, and WINS server address assignments. It is not recommended to use L2TP and XAuth together for an L2TP-over-IPSec tunnel. Not only do the two protocols accomplish the same goals, but the L2TP address assignments overwrite the XAuth address assignments after Phase 2 IKE negotiations complete and L2TP negotiations take place.

- If you create a multiple-type user account on the local database combining auth/L2TP or auth/XAuth, the same username and password must be used for both logins.

Although it is more convenient to create a single multiple-type user account, separating the user types into two single accounts allows you to increase security. For example, you can store an auth user account on an external auth server and an XAuth user account on the local database. You can then assign different login usernames and passwords to each account and reference the XAuth user in the IKE gateway configuration and the auth user in the policy configuration. The dialup VPN user must authenticate himself twice, potentially with two completely different usernames and passwords.

Group Expressions

A group expression is a statement that you can use in policies to conditionalize the requirements for authentication. Group expressions allow you to combine users, user groups, or other group expressions as alternatives for authentication (“a” OR “b”), or as requirements for authentication (“a” AND “b”). You can also use group expressions to exclude a user, user group, or another group expression (NOT “c”).

NOTE: Although you define group expressions on the security device (and store them on the local database), the users and user groups that you reference in the group expressions must be stored on an external RADIUS server. A RADIUS server allows a user to belong to more than one user group. The local database does not permit this.

Group expressions make use of the three operators OR, AND, and NOT. The objects in the expression to which OR, AND, and NOT relate can be an auth user, an auth user group, or a previously defined group expression. Table 1 lists objects, group expressions, and examples.

Table 1: Group Expression Examples (page 1 of 2)

Object	Expression	Example
Users	OR	A policy specifies that the user be <i>a</i> OR <i>b</i> , so the security device authenticates if the user matches either condition <i>a</i> or <i>b</i> .
	AND	AND in a group expression requires that at least one of the two expression objects be either a user group or a group expression. (It is illogical to require a user to be user <i>a</i> AND user <i>b</i> .) If the authentication aspect of a policy requires that the user be <i>a</i> AND a member of group <i>b</i> , then the security device authenticates the user only if those two conditions are met.
	NOT	A policy specifies that the user be anyone except user <i>c</i> (NOT <i>c</i>), then the security device authenticates as long as the user is not <i>c</i> .

Table 1: Group Expression Examples (page 2 of 2)

Object	Expression	Example
User groups	OR	A policy specifies that the user belong to group <i>a</i> OR group <i>b</i> , so the security device authenticates if the user belongs to either group.
	AND	A policy requires that the user belong to group <i>a</i> AND group <i>b</i> , so the security device authenticates the user only if he or she belongs to both groups.
	NOT	A policy specifies that the user belong to any group other than group “ <i>c</i> ” (NOT “ <i>c</i> ”), so the security device authenticates the user as long as the user does not belong to that group.
Group expressions	OR	A policy specifies that the user fit the description of group expression <i>a</i> OR group expression <i>b</i> , so the security device authenticates the user if either group expression applies.
	AND	A policy specifies that the user fit the description of group expression <i>a</i> AND group expression <i>b</i> , so the security device allows authentication only if both group expressions apply to the user.
	NOT	A policy specifies that the user not fit the description of group expression <i>c</i> (NOT <i>c</i>), so the security device allows authentication only if the user does not fit that group expression.

Example: Group Expressions (AND)

In this example, you create a group expression “s+ m” that states “sales AND marketing”. You have previously created the auth user groups “sales” and “marketing” on an external RADIUS auth server named “radius1” and populated them with users. (For an example on how to configure an external RADIUS auth server, see “Example: RADIUS Auth Server” on page 33.) You then use that group expression in an intrazone policy whose authentication component requires a user be a member of both user groups to be able to access the confidential contents on a server named “project1” (10.1.1.70).

NOTE: For an intrazone policy to work properly, the source and destination addresses must be in different subnets connected to the security device through interfaces that are both bound to the same zone. There cannot be any other routing device beside the security device that can route traffic between the two addresses. For more information about intrazone policies, see “Policies” on page 2-159.

WebUI**1. Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: project1
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.1.70/32
 Zone: Trust

2. Group Expression

Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

Group Expression: s+m
 AND: (select), sales AND marketing

3. Policy

Policy > Policies > (From: Trust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), project1
 Service: ANY
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: radius1
 Group Expression: (select), External Group Expression - s+m

CLI**1. Address**

```
set address trust project1 10.1.1.70/32
```

2. Group Expression

```
set group-expression s+m sales and marketing
```

3. Policy

```
set policy top from trust to trust any project1 any permit auth server radius1
  group-expression s+m
save
```

Example: Group Expressions (OR)

In this example, you create a group expression “a/b” that states “amy OR basil”. You have previously created auth user accounts “amy” and “basil” on an external RADIUS auth server named “radius1.” (For an example on how to configure an external RADIUS auth server, see “Example: RADIUS Auth Server” on page 33.) You then use that group expression in a policy from the Trust zone to the DMZ. The authentication component of the policy requires the user to be either amy or basil to be able to access the webserver named “web1” at 210.1.1.70.

WebUI

1. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: web1
 IP Address/Domain Name
 IP/Netmask: (select), 210.1.1.70/32
 Zone: DMZ

2. Group Expression

Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

Group Expression: a/b
 OR: (select), amy OR basil

3. Policy

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), web1
 Service: ANY
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: radius1
 Group Expression: (select), External Group Expression - a/b

CLI

1. Address

```
set address trust project1 210.1.1.70/32
```

2. Group Expression

```
set group-expression a/b amy or basil
```

3. Policy

```
set policy top from trust to dmz any web1 any permit auth server radius1
  group-expression a/b
save
```

Example: Group Expressions (NOT)

In this example, you create a group expression “-temp” that states “NOT temp”. You have previously created a local auth user group “temp” on an external RADIUS auth server named “radius1.” (For an example on how to configure an external RADIUS auth server, see “Example: RADIUS Auth Server” on page 33.) You then use that group expression in a policy from the Trust zone to the Untrust zone that allows Internet access to all full-time employees, but not to temporary contractors. The authentication component of the policy requires everyone in the Trust zone to be authenticated except the users in “temp,” who are denied access to the Untrust zone.

WebUI**1. Group Expression**

Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

```
Group Expression: -temp
OR: (select), NOT temp
```

2. Policy

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

```
Source Address:
  Address Book Entry: (select), Any
Destination Address:
  Address Book Entry: (select), Any
Service: HTTP
Action: Permit
Position at Top: (select)
```

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

```
Authentication: (select)
Auth Server: (select)
Use: Local
Group Expression: (select), External Group Expression - -temp
```

CLI**1. Group Expression**

```
set group-expression -temp not temp
```

2. Policy

```
set policy top from trust to untrust any any any permit auth server radius1
  group-expression -temp
save
```

Banner Customization

A banner is a message that appears on a monitor in different places depending on the type of login:

- At the top of a Telnet or console display when an admin user connects to the security device

NOTE: You can include an additional banner line under a Telnet or console banner. The second banner line remains the same for both Telnet and console login displays although the Telnet banner can differ from the console banner. To create a secondary banner, enter the following command: **set admin auth banner secondary string**.

- At the top of a browser screen after an auth user has successfully logged into a WebAuth address
- Before or after a Telnet, an FTP, or an HTTP login prompt, success message, and fail message for auth users

All of the banners, except that for a console login, already have default messages. You can customize the messages that appear on the banners to better suit the network environment in which you use the security device.

Example: Customizing a WebAuth Banner

In this example, you change the message that appears in the browser to indicate that an auth user has successfully authenticated himself after successfully logging in via WebAuth. The new message is “Authentication approved.”

WebUI

Configuration > Admin > Banners > WebAuth: In the Success Banner field, type **Authentication approved**, then click **Apply**.

CLI

```
set webauth banner success "Authentication approved"
save
```

Login Banner

The size of the login banner is increased to a maximum of 4Kbytes. This provides space for terms of use statements, which are presented before administrators and authenticated users log into the security device and into protected resources behind the device. The login banner is a clear text ASCII file you create and store on the security device, the file must be called **usrterms.txt**. You activate the banner by restarting of the system. If the banner file is greater than 4Kbytes, the security device will not accept it and will continue using existing banners entered through the CLI and the WebUI.

When activated, the login banner is used globally by the root system and all virtual systems (vsys). You cannot differentiate or customize between or within a vsys. The login banner pre-empts all individually defined administrative access banners and firewall authentication banners. After entering a username and password, the user must click the **Login** button. Pressing the **Enter** key will not log the user into the device.

Example: Creating a Login Banner

Use the SCP utility to securely copy the banner file to the security device. With the following command, an administrator with username **netscreen** copies the banner file **my_large_banner.txt** to a security device at IP address 1.1.1.2. The banner file must be saved on the security device as **usrterms.txt**.

```
linux: ~#scp my_large_banner.txt netscreen@1.1.1.2:useterms.txt
```

You must restart the device to activate the new banner. To modify the banner file, create a new file and overwrite the existing one with the new one.

To remove the banner, issue the following command on the security device:

```
device-> delete file usrterms.txt
```

This disables the login banner feature after you restart the device.

Chapter 2

Authentication Servers

This chapter examines different kinds of authentication servers—the local database built into every security device, and external RADIUS, SecurID, and LDAP authentication servers. This chapter includes the following sections:

- “Authentication Server Types” on page 13
- “Local Database” on page 15
- “External Authentication Servers” on page 17
- “Auth Server Types” on page 19
 - “Remote Authentication Dial-In User Service” on page 19
 - “SecurID” on page 27
 - “Lightweight Directory Access Protocol” on page 29
 - Terminal Access Control Access Control System Plus (TACACS+) on page 30
- “Prioritizing Admin Authentication” on page 32
- “Defining Auth Server Objects” on page 33
- “Defining Default Auth Servers” on page 39

Authentication Server Types

You can configure the security device to use the local database or one or more external authentication servers to verify the identities of the following types of users:

- Auth
- IKE
- L2TP
- XAuth

- Admin
- 802.1x

NOTE: IKE user accounts must be stored on the local database. The only external server to support L2TP and XAuth remote setting assignments and admin privilege assignments is RADIUS.

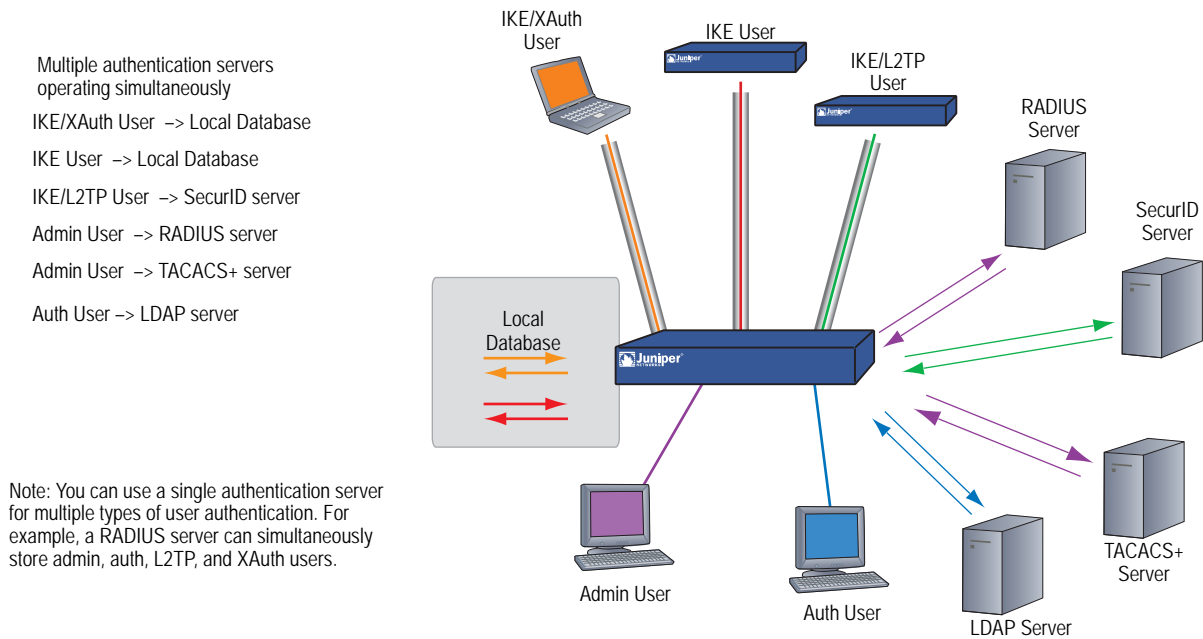
In addition to its local database, a security device supports external RADIUS, SecurID, LDAP, and TACACS+ servers. You can use each kind of authentication server to authenticate auth users, L2TP users, XAuth users, and admin users. ScreenOS also supports WebAuth, an alternative authentication scheme for auth users. (For a WebAuth example, see “Example: WebAuth + SSL Only (External User Group)” on page 63.) Any auth server that contains auth user account types is eligible to be the default WebAuth auth server. Table 2 lists supported servers types and authentication features.

Table 2: Authentication Server Type, User Types, and Features

Server Type	Supported User Types and Features									
	Auth Users	IKE Users	L2TP Users		XAuth Users		Admin Users		User Groups	Group Expressions
			Auth	Remote Settings	Auth	Remote Settings	Auth	Privileges		
Local	X	X	X	X	X	X	X	X	X	
RADIUS	X		X	X	X	X	X	X	X	X
SecurID	X		X		X		X			
LDAP	X		X		X		X			
TACACS+							X	X		

On most Juniper Networks security devices, you can simultaneously employ up to 10 primary authentication servers per system—root system and virtual system—in any combination of types. This total includes the local database and excludes backup authentication servers. A RADIUS or LDAP server supports two backup servers, and a SecurID server supports one backup server; so, for example, you might use the local database and nine different primary RADIUS servers, with each RADIUS server having two backup servers assigned to it. See Figure 4.

Figure 4: Types of Authentication Servers

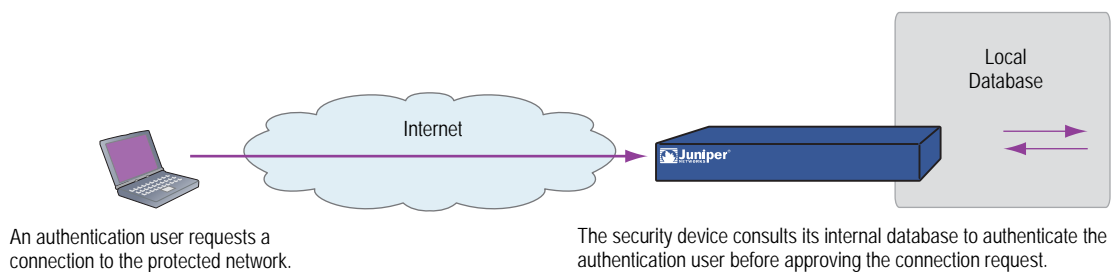


The following sections explain the local database and each authentication server in detail.

Local Database

All Juniper Networks security devices support a built-in user database for authentication. When you define a user on the security device, the security device enters the username and password in its local database. See Figure 5.

Figure 5: Local Authentication



The local database supports the following types of users and authentication features:

- Users:
 - Auth
 - IKE
 - L2TP
 - XAuth
 - Admin
 - 802.1x
- Authentication features:
 - Admin privileges
 - WebAuth
 - User groups
 - Group expressions

NOTE: You define the group expressions on the security device, but the users and user groups must be stored on an external RADIUS auth server. For more information about group expressions, see “Group Expressions” on page 5.

The local database is the default authentication server (auth server) for all types of authentication. For instructions on how to add users and user groups to the local database via the WebUI and CLI, see “Authentication Users” on page 47 and “IKE, XAuth, and L2TP Users” on page 67.

Example: Local Database Timeout

By default, the local database authentication timeout for both admins and auth users is 10 minutes. In this example, you change it to never time out for admins and to time out after 30 minutes for auth users.

WebUI

Configuration > Admin > Management: Clear the Enable Web Management Idle Timeout check box, then click **Apply**.

Configuration > Auth > Servers > Edit (for Local): Enter **30** in the Timeout field, then click **Apply**.

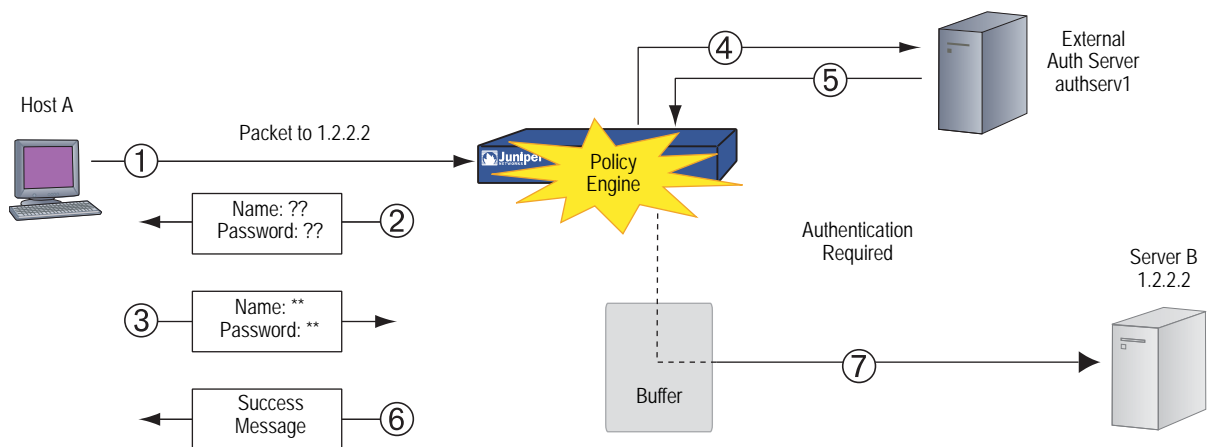
CLI

```
set admin auth timeout 0
set auth-server Local timeout 30
save
```

External Authentication Servers

A security device can connect to one or more external authentication servers, or *auth servers*, on which you store user accounts. When the security device receives a connection request that requires authentication verification, the security device requests an authentication check from the external auth server specified in the policy, L2TP tunnel configuration, or IKE gateway configuration. The security device then acts as a relay between the user requesting authentication and the auth server granting authentication. Figure 6 shows the steps to a successful authentication check by an external auth server.

Figure 6: External Auth Server



1. Host A sends an FTP, an HTTP, or a Telnet TCP SYN packet to 1.2.2.2.
2. The security device intercepts the packet, notes that its corresponding policy requires authentication from authserv1, buffers the packet, and prompts the user for a username and password.
3. The user replies with a username and password.
4. The security device relays the login information to authserv1.
5. Authserv1 sends back a notification of success to the security device.
6. The security device informs the auth user of his or her login success.
7. The security device then forwards the packet from its buffer to its destination of 1.2.2.2.

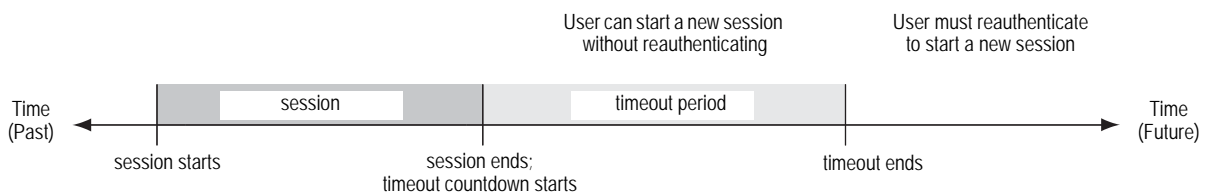
Auth Server Object Properties

A security device treats each auth server as an object that it can reference in policies, IKE gateways, and L2TP tunnels. The properties described in Table 3 define and uniquely identify an auth server object.

Table 3: Auth Server Object Properties

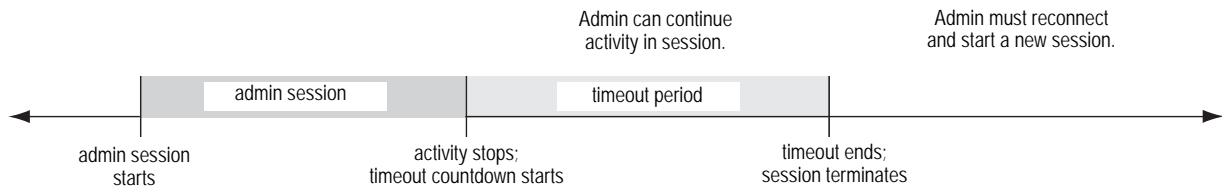
Property	Description
Object name	A name string, such as authserv1. (The only predefined auth server is Local.)
ID number	You can set the ID number or allow the security device to set it automatically. If you set an ID number, you must choose one that is not already in use.
Type	RADIUS, SecurID, LDAP, TACACS+ .
Server name	The IP address or domain name of the server.
Backup1	The IP address or domain name of a primary backup server.
Backup2	The IP address or domain name of a secondary backup server.
Account Type	One or more of the following types of users: Auth, L2TP, 802.1x, XAuth; or Admin by itself.
Timeout value	The timeout value is idle timeout, and takes on a different meaning if it is for an auth user or if it is for an admin user.
Auth user	The timeout countdown begins after the first authenticated session completes. If the user initiates a new session before the countdown reaches the timeout threshold, the timeout countdown resets. The default timeout value is 10 minutes, the maximum is 255 minutes. To disable the timeout feature, set the timeout value to 0. See Figure 7.
Admin user	If the length of idle time reaches the timeout threshold, the security device terminates the admin session. To continue managing the security device, the admin must reconnect to the device and reauthenticate himself. The default timeout value is 10 minutes, the maximum is 1000 minutes. To disable the timeout feature, set the timeout value to 0. See Figure 8.
Forced Timeout	Forced timeout, unlike idle timeout, does not depend on the idleness of the user, but on an absolute timeout after which access for the authenticated user is terminated. The auth table entry for the user is removed, as are all associated sessions for the auth table entry. The default is 0 (disabled), the range is 0 to 10000 (6.9 days).

Figure 7: Auth Server Object Properties



NOTE: User authentication timeout is not the same as session idle timeout. If no activity occurs in a session for a predefined length of time, the security device automatically removes the session from its session table.

Figure 8: Admin Timeout Property



In addition to the above properties that apply to all auth server objects, each server has a few others specific to itself. These are explained in “Auth Server Types.”

Auth Server Types

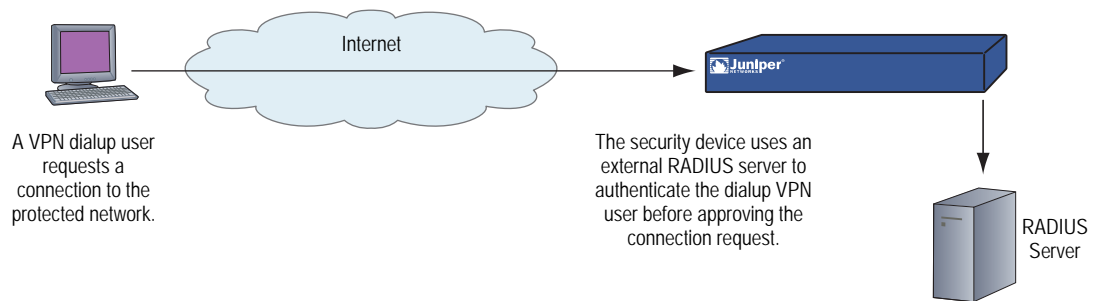
In addition to the internal database, security devices support four types of external auth servers:

- Remote Authentication Dial-In User Service (RADIUS)
- SecurID
- Light-Weight Directory Access Protocol (LDAP)
- Terminal Access Controller Access Control System Plus (TACACS+)

Remote Authentication Dial-In User Service

The Remote Authentication Dial-In User Service (RADIUS) is a protocol for an authentication server that can support up to tens of thousands of users.

Figure 9: Using RADIUS as an External Auth Server



The RADIUS client (that is, the security device) authenticates users through a series of communications between the client and the server. Basically, RADIUS asks the person logging in to enter his or her username and password. It then compares these values to those in its database, and once a user is authenticated, the client provides the user with access to the appropriate network services.

To configure the security device for RADIUS, you must specify the IP address of the RADIUS server and define a shared secret—the same as that defined on the RADIUS server. The shared secret is a password the RADIUS server uses to generate a key to encrypt traffic between the security and RADIUS devices.

RADIUS Auth Server Object Properties

In addition to the generic auth server properties listed in “Auth Server Object Properties” on page 18, a RADIUS server also makes use of the properties described in Table 4.

Table 4: Radius Auth Server Object Properties

Property	Description
Shared Secret	The secret (password) shared between the security device and the RADIUS server. The devices use this secret to encrypt the user’s password that it sends to the RADIUS server.
RADIUS Port	The port number on the RADIUS server to which the security device sends authentication requests. The default port number is 1645.
RADIUS Retry Timeout	The interval (in seconds) that the security device waits before sending another authentication request to the RADIUS server if the previous request does not elicit a response. The default is three seconds.

Supported User Types and Features

A RADIUS server supports the following types of users and authentication features:

- Auth users
- L2TP users (authentication and remote settings)
- Admin users (authentication and privilege assignments)
- User groups
- XAuth users (authentication and remote settings)

The XAuth module provides support for the Session-timeout and Idle-timeout attributes retrieved from the RADIUS server described in Table 5.

Table 5: XAuth Attribute Support (page 1 of 2)

Attribute	Description
Session-timeout	If the Session-timeout attribute is a non-zero value, the phase-1/phase-2 security association (SA) and the XAuth are both terminated when the timeout value is reached.
Idle-timeout	If the Idle-timeout attribute is a non-zero value, it takes preference over the local phase-2 SA idle time configuration and member SA hold time. If the Idle-timeout value is 0, the local phase-2 SA idle time and member SA hold time is used.
XAuth Accounting Start	The XAuth Accounting start is sent to the external RADIUS server after the user is authenticated correctly.

Table 5: XAuth Attribute Support (page 2 of 2)

Attribute	Description
XAuth Accounting Stop	<p>The XAuth Accounting stop is sent to the external RADIUS server when the XAuth connection is torn down. All phase-1/phase-2 SA and XAuth connection is terminated under the following conditions:</p> <ul style="list-style-type: none"> ■ RADIUS server Session-timeout attribute is reached ■ RADIUS server Session-timeout attribute is not configured The XAuth session lifetime is used instead. ■ RADIUS server Idle-timeout attribute is reached on all Phase-2 SAs ■ Client disconnect is detected via dead peer detection (DPD) or heartbeat. ■ Locally configured phase-2 SA idle time or the member SA hold time is reached because RADIUS server is not providing an Idle-timeout attribute.

A RADIUS server can support all of the user types and features that the local database supports except IKE users. Among the four types of external auth servers, RADIUS is the only one at this time with such broad support. For a RADIUS server to support such ScreenOS-specific attributes as admin privileges, user groups, and remote L2TP and XAuth IP address, and DNS and WINS server address assignments, you must load a RADIUS dictionary file that defines these attributes onto the RADIUS server.

NOTE: ScreenOS uses the standard RADIUS attribute for IP address assignments. If you only want to use RADIUS for IP address assignments, you do not have to load the ScreenOS vendor-specific attributes (VSAs).

RADIUS Dictionary File

A dictionary file defines vendor-specific attributes (VSAs) that you can load onto a RADIUS server. After defining values for these VSAs, ScreenOS can then query them when a user logs into a security device. ScreenOS VSAs include admin privileges, user groups, and remote L2TP and XAuth IP address, and DNS and WINS server address assignments. There are two RADIUS dictionary files, one for Cisco RADIUS servers and one for Funk Software RADIUS servers. If you are using a Microsoft RADIUS server, there is no dictionary file. You must configure it as outlined in *Bi-Directional Remote VPN using xAuth and Firewall Authentication with Microsoft Internet Authentication Service (IAS)*, which you can download from <http://kb.juniper.net/kb/documents/public/kbdocs/ns10382/ns10382.pdf>

Each RADIUS dictionary file contains the specific information described in Table 6.

Table 6: RADIUS Dictionary File Contents

Field	Description
Vendor ID	The ScreenOS vendor ID (VID; also called an “IETF number”) is 3224. The VID identifies a specific vendor for a particular attribute. Some types of RADIUS server require you to enter the VID for each attribute entry, while other types only require you to enter it once and then apply it globally. Refer to your RADIUS server documentation for further information.
Attribute Name	The attribute names describe individual ScreenOS-specific attributes, such as NS-Admin-Privilege, NS-User-Group, NS-Primary-DNS-Server, and so on.
Attribute Number	The attribute number identifies an individual vendor-specific attribute. ScreenOS-specific attribute numbers fall into two ranges: <ul style="list-style-type: none"> ■ ScreenOS: 1 – 199 ■ Global PRO: 200 and above For example, the ScreenOS attribute number for user groups is 3. The Global PRO attribute number for user groups is 200.
Attribute Type	The attribute type identifies the form in which attribute data (or “value”) appears—a string, an IP address, or an integer.

The RADIUS server automatically receives the above information when you load the RADIUS dictionary file onto it. To make new data entries, you must manually enter a value in the form indicated by the attribute type. For example, an entry for a read-write admin appears as follows:

VID	Attribute Name	Attribute Number	Attribute Type	Value
3224	NS-Admin-Privileges	1	data= int4 (i.e., integer)	2 (2 = all privileges)

To download a dictionary file, go to http://www.juniper.net/customers/csc/research/netscreen_kb/downloads/dictionary/funk_radius.zip

or

http://www.juniper.net/customers/csc/research/netscreen_kb/downloads/dictionary/cisco_radius.zip

Log in and save the file to a local drive.

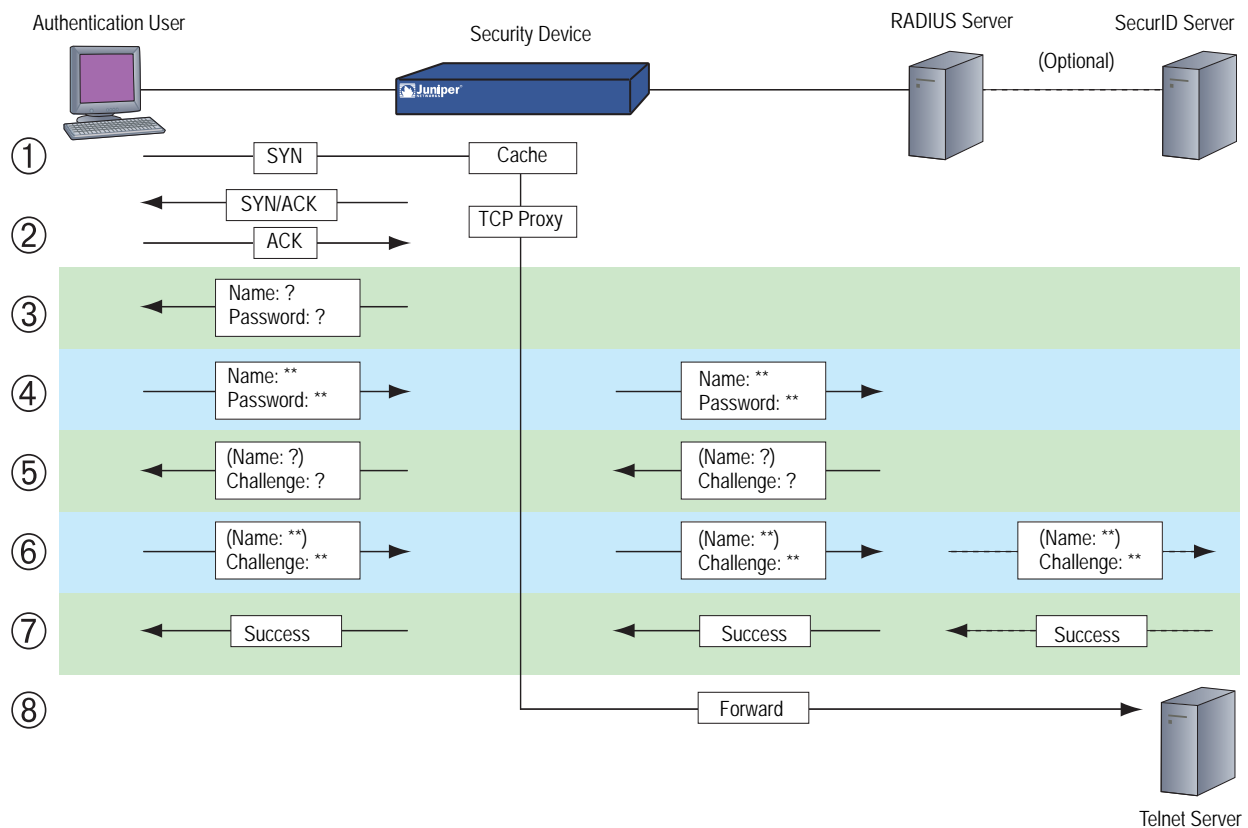
NOTE: All new installations of Funk Steel Belted RADIUS has the RADIUS firewall dictionary file already loaded on the RADIUS server.

RADIUS Access Challenge

Juniper Networks security devices can now process access-challenge packets from an external RADIUS server when an authentication user attempts to log in via Telnet. Access challenge presents an additional condition to the login process after the approval of a username and password. After an authentication user responds to a login prompt with the correct username and password, the RADIUS server sends an access challenge to the security device, which then forwards it to the user. When

the user replies, the security device sends a new access request with the user's response to the RADIUS server. If the user's response is correct, the authentication process concludes successfully. Figure 10 lists the steps required for an authentication user who wants to telnet to a server.

Figure 10: RADIUS Access-Challenge Sequence



1. An authentication user sends a SYN packet to initiate a TCP connection for a Telnet session to a Telnet server.
2. A security device intercepts the packet, checks its policy list, and determines that this session requires user authentication. The security device caches the SYN packet and proxies the TCP 3-way handshake with the user.
3. The security device prompts the user to log in with a username and password.
4. The authentication user enters his or her username and password and sends it to the security device. The security device then sends an access request with the login information to a RADIUS server.
5. If the information is correct, the RADIUS server sends the security device an access challenge with a reply-message attribute that prompts the user to provide a response to a challenge. (The access challenge can optionally prompt the authentication to provide a username again. The second username can be the same as the first or a different one.) The security device then sends the user another login prompt that contains the content of the reply-message attribute.

6. The authentication user enters his or her challenge response (and, optionally, a username) and sends it to the security device. The security device then sends a second access request, with the user's challenge response, to the RADIUS server.

If the RADIUS server needs to authenticate the challenge response via another auth server—for example, if a SecurID server must authenticate a token code—the RADIUS server sends the access request to the other auth server.

7. If the RADIUS server forwarded the challenge response to another auth server and that server sends an access accept, or, if the RADIUS server itself approves the challenge response, the RADIUS server sends an access-accept message to the security device. The security device then notifies the authentication user that his or her login is successful.
8. The security device forwards the initial SYN packet to its original destination: the Telnet server.

NOTE: ScreenOS does not support access challenge with L2TP at the time of this release.

Supported RADIUS Enhancements for Auth and XAuth Users

ScreenOS supports RADIUS enhancements through the Authentication and Extended Authentication (XAuth) modules with the following attributes:

- NS Access Service Type on page 24
- Framed Pool and Framed IP Address on page 25
- Account Session ID on page 25
- Calling Station ID on page 26
- Called Station ID on page 26
- Compatibility RFC-2138 on page 26
- Username on page 26
- Separator on page 26
- Fail-over on page 27

NS Access Service Type

The **NS-Access-Service-Type** attribute provides information about the service type. The security device adds this attribute to each Access-Request indicating the type of service required by the user. This attribute is enabled by default.

If the RADIUS module receives the request from a Telnet, FTP, or HTTP Authentication module, it sets the value to WEB-AUTH (2). If the RADIUS module receives the request from the XAuth module, it sets the value to VPN-IPSEC (3).

The device includes the *ns-access-service-type* and value in the Access-Request message. If the RADIUS server determines that the requesting user is allowed to access to the service, it sends an Access-Accept message. If the service-type is not applicable to the requesting user, the RADIUS server sends an Access-Reject message.

The RADIUS server does not include the *ns-access-service-type* attribute in the Access-Response messages.

Framed Pool and Framed IP Address

The RADIUS server includes the **framed-pool** attribute in the Access-Accept message. When the Framed-Pool attribute is included, the device allocates an IP address to the user from this pool. However, the device does not send the Framed-Pool attribute in Access-Request messages.

Table 7 shows how the device handles the framed-pool and framed-ip-address attributes. The RADIUS enhancements also includes the ability to handle address pools at the virtual system (VSYS) level.

Table 7: Supported Attributes

Supported Attributes	Resolution
Framed-Pool attribute and the Framed-IP-Address attribute are both included in the Access-Accept message.	The Framed-Pool attribute is always ignored by the RADIUS server unless the Framed-IP-Address value is 0xFFFFFFFF (255.255.255.254). Then, the device allocates an address from the Framed-Pool attribute sent by the RADIUS server.
Framed-Pool attribute and the Framed-IP-Address attribute are both absent from the Access-Accept message.	The device does not assign an IP address to the end user.
Framed-IP-Address attribute is included in the Access-Accept message and it has a value of 0xFFFFFFFF (255.255.255.254). Framed-Pool attribute is absent.	The device allocates an IP address from the default IP address pool that is configured for that VSYS.
The pool sent out in the Framed-Pool attribute is not configured, or it does not have any IP addresses.	<p>The following error messages are generated and the negotiation is terminated:</p> <ul style="list-style-type: none"> ■ Login failed: IP pool needed but not configured. ■ Login failed: No more IP address available in IP pool. <p>In both scenarios, the client receives the following message:</p> <p>No more IP address available in IP pool</p>

Account Session ID

The **acct-session-id** uniquely identifies the accounting session. Each time an XAuth user connects to the device and the device authenticates the user, the device establishes a new acct-session-id, which identifies the accounting session. The accounting session lasts between the time the device sends the RADIUS server an Accounting-Start message, and the time it sends an Accounting-Stop message. To identify the user, each RADIUS access or request message may contain the calling-station-id (described below).

The **acct-session-id length** *number* is the length of the account-session-id in bytes. The default length of this value is 11 bytes. The number setting is for accommodating some RADIUS servers, which may have problems with the default length. You can set the length of acct-session-id from 6 bytes to 10 bytes, inclusive. To restore the default setting, execute the following command:

```
unset auth-server name_str radius attribute acct-session-id number
```

Calling Station ID

The calling-station-id attribute identifies the originator of the call. For example, this value might consist of the phone number of the user originating the call.

Called Station ID

The called-station-id attribute identifies the destination or receiver of the call. For example, this value might consist of the phone number of the user originating the call.

Compatibility RFC-2138

The **compatibility rfc-2138** attribute makes RADIUS accounting comply with RFC 2138, as compared with RFC 2865. For operations where RFC 2865 (the most recent standard) and RFC 2138 are mutually exclusive, the command works in accordance with RFC 2138, instead of RFC 2865. In cases where the behavior is additive, the command works compatibly with both RFC 2865 and RFC 2138.

Username

The **username** specifies a domain name for a particular auth server, or a portion of a username from which to strip characters. If you specify a domain name for the auth server, it must be present in the username during authentication.

Separator

The device uses a separator character to identify where stripping occurs. Stripping removes all characters to the right of each instance of the specified character, plus the character itself. The device starts with the right most separator character. An example of a separator command is as follows:

```
set auth-server name_str username separator string number number
```

where:

- *name_str* is the name of the authentication server.
- *string* is the character separator.
- *number* is the number of character separator instances with which to perform the character stripping.

If the specified number of separator characters (*number*) exceeds the actual number of separator characters in the username, the command stops stripping at the last available separator character.

NOTE: The device performs domain-name matching before stripping.

Fail-over

The **fail-over** attribute specifies the revert-interval (expressed in seconds) that must pass after an authentication attempt, before the device attempts authentication through backup authentication servers.

When an authentication request sent to a primary server fails, the device tries the backup servers. If authentication via a backup server is successful, and the revert-interval time interval has elapsed, the device sends subsequent authentication requests to the backup server. Otherwise, it resumes sending the requests to the primary server. The range is 0 seconds (disabled) to 86400 seconds.

An example of the fail-over and revert-interval command is as follows:

```
set auth-server name_str fail-over revert-interval number
```

where:

- *name_str* is the name of the authentication server.
- *number* is the length of time (expressed in seconds).

NOTE: This feature applies to RADIUS and LDAP servers only.

SecurID

Instead of a fixed password, SecurID combines two factors to create a dynamically changing password. SecurID issues a credit-card-sized device, known as an *authenticator*, with an LCD window that displays a randomly generated string of numbers (a *token code*) that changes every minute. The user also has a personal identification number (PIN). When the user logs on, he enters a username and his PIN plus the current token code. See Figure 11.

Figure 11: SecurID Token



The authenticator performs an algorithm known only by RSA to create the values that appear in the LCD window. When the user to be authenticated enters his PIN and the number on his card, the ACE server, which also performs the same algorithm, compares the values received with those in its database. If they match, the authentication is successful.

The relationship of security device and a RSA SecurID ACE server is similar to that of a security device and a RADIUS server. That is, the security device acts as a client, forwarding authentication requests to the external server for approval and relaying login information between the user and the server. SecurID differs from RADIUS in that the user’s “password” involves a continually changing token code.

SecurID Auth Server Object Properties

In addition to the generic auth server properties listed in “Auth Server Object Properties” on page 18, a SecurID server also makes use of the properties described in Table 8.

Table 8: SecurID Auth Server Object Properties

Property	Description
Authentication Port	The port number on the SecurID ACE server to which the security device sends authentication requests. The default port number is 5500.
Encryption Type	The algorithm used for encrypting communication between the security device and the SecurID ACE server—either SDI or DES.
Client Retries	The number of times that the SecurID client (that is, the security device) tries to establish communication with the SecurID ACE server before aborting the attempt.
Client Timeout	The length of time in seconds that the security device waits between authentication retry attempts.
Use Duress	An option that prevents or allows use of a different PIN number. When this option is enabled, and a user enters a previously determined duress PIN number, the security device sends a signal to the SecurID ACE server, indicating that the user is performing the login against his or her will; that is, while under duress. The SecurID ACE server permits access that one time, and then it denies any further login attempts by that user until he or she contacts the SecurID administrator. Duress mode is available only if the SecurID ACE server supports this option.

Supported User Types and Features

A SecurID ACE server supports the following types of users and authentication features:

- Auth users
- L2TP users (user authentication; L2TP user receives default L2TP settings from the security device)
- XAuth users (user authentication; no support for remote setting assignments)
- Admin users (user authentication; admin user receives default privilege assignment of read-only)

At present, a SecurID ACE server cannot assign L2TP or XAuth remote settings or ScreenOS admin privileges, although you can use a SecurID server to store L2TP, XAuth, and admin user accounts for authentication purposes. Also, ScreenOS does not provide user group support when used with SecurID.

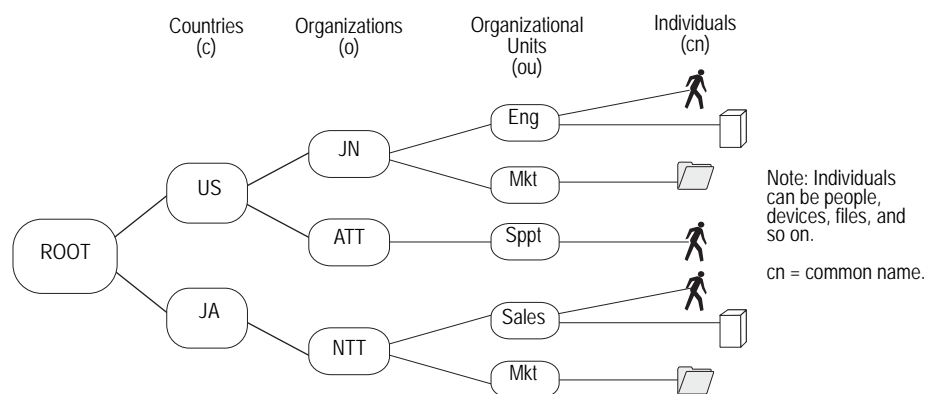
Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a directory server standard developed at the University of Michigan in 1996. LDAP is a protocol for organizing and accessing information in a hierarchical structure resembling a branching tree. Its purpose is twofold:

- To locate resources, such as organizations, individuals, and files on a network
- To help authenticate users attempting to connect to networks controlled by directory servers

The basic LDAP structure branches from countries to organizations to organizational units to individuals. There can also be other, intermediary levels of branching, such as “states” and “counties.” Figure 12 shows an example of the branching organizational structure of LDAP.

Figure 12: LDAP Hierarchical Structure



NOTE: For information about LDAP, refer to RFC 1777, *Lightweight Directory Access Protocol*.

You can configure the security device to link to a Lightweight Directory Access Protocol (LDAP) server. This server uses the LDAP hierarchical syntax to identify each user uniquely.

LDAP Auth Server Object Properties

In addition to the generic auth server properties listed in “Auth Server Object Properties” on page 18, an LDAP server also makes use of the properties described in Table 9.

Table 9: LDAP Auth Server Object Properties

Property	Description
LDAP Server Port	The port number on the LDAP server to which the security device sends authentication requests. The default port number is 389. Note: If you change the LDAP port number on the security device, also change it on the LDAP server.
Common Name Identifier	The identifier used by the LDAP server to identify the individual entered in a LDAP server. For example, an entry of “uid” means “user ID” and “cn” for “common name.”
Distinguished Name (dn)	The path used by the LDAP server before using the common name identifier to search for a specific entry. (For example, c= us;o= juniper, where “c” stands for “country” and “o” for “organization.”)

Supported User Types and Features

An LDAP server supports the following types of users and authentication features:

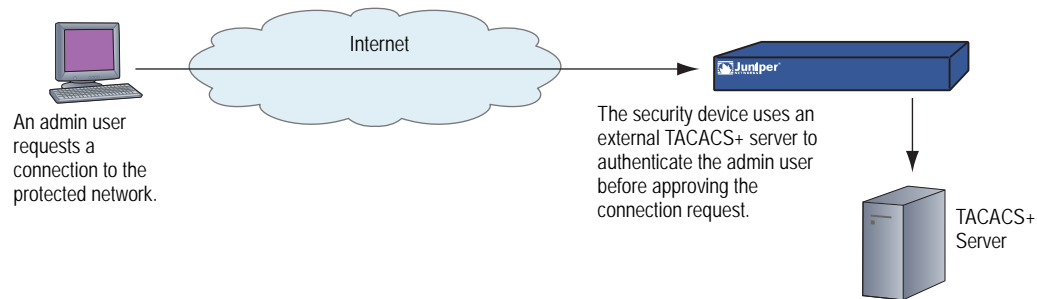
- Auth users
- L2TP users (user authentication; L2TP user receives default L2TP settings from the security device)
- XAuth users (user authentication; no support for remote setting assignments)
- Admin users (user authentication; admin user receives default privilege assignment of read-only)

At present, an LDAP server cannot assign L2TP or XAuth remote settings or ScreenOS admin privileges, although you can use an LDAP server to store L2TP, XAuth, and admin user accounts for authentication purposes. Also, ScreenOS does not provide user group support when used with LDAP.

Terminal Access Control Access Control System Plus (TACACS+)

Terminal Access Controller Access Control System Plus (TACACS+) is a security application that provides centralized validation of users. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation.

Figure 13: Authenticating to a TACACS+ Server



The TACACS+ client (that is, the security device) authenticates users through a series of communications between the client and the server. Basically, TACACS+ asks the person logging in to enter his or her username and password. It then compares these values to those in its database, and once a user is authenticated, the client provides the user with access to the appropriate network services.

To configure the security device for TACACS+, you must specify the IP address of the TACACS+ server and define a shared secret—the same as that defined on the TACACS+ server. The shared secret is a password the TACACS+ server uses to generate a key to encrypt traffic between the security and TACACS+ devices.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. The TACACS+ support in ScreenOS allows the TACACS+ server to provide authentication and authorization independently as follows:

- Authentication and authorization

TACACS+ separates authentication and authorization functions. Remote authentication is supported for admin users only. Authenticated administrators are assigned a privilege level and vsys. ScreenOS supports user-level authorization only.

- Configure up to two TACACS+ servers

If the primary TACACS+ server is not reachable, then the backup TACACS+ server is queried. For more information, see "Prioritizing Admin Authentication" on page 32.

TACACS+Server Object Properties

In addition to the generic auth server properties listed in “Auth Server Object Properties” on page 18, a TACACS+ server also makes use of the properties described in Table 10.

Table 10: TACACS+Server Object Properties

Property	Description
Shared Secret	The secret (password) shared between the security device and the TACACS+ server. The devices use this secret to encrypt the user’s password that it sends to the TACACS+ server.
TACACS+ Port	The port number on the TACACS+ server to which the security device sends authentication requests. The default port number is 49.
TACACS+ Retry Timeout	The interval (in seconds) that the security device waits before sending another authentication request to the TACACS+ server if the previous request does not elicit a response. The default is three seconds.
Client Retries	The number of times that the TACACS+ client (that is, the security device) tries to establish communication with the TACACS+ server before aborting the attempt.
Client Timeout	The length of time in seconds that the security device waits between authentication retry attempts.
Encryption	TACACS+ encrypts the entire payload of the client server exchange.

Prioritizing Admin Authentication

ScreenOS lets you prioritize the authentication process with regards to local and remote authentication services. The admin defines the sequence in which the authentication service is queried and the action to take if the initial attempt fails.

- Assigning priorities to authentication services

Set higher priority to authenticate to the remote auth server over the local database. Root-privileged admins can define the sequence of the authentication service (local and remote) in which admin authentication service is attempted. The root admin sets one of the authentication services as primary. The other automatically becomes a secondary service.

- Defining fallback behavior

If the primary authentication service fails, you can configure the device to authenticate to the secondary service (default) or bypass it. The above action is defined differently for root-privileged and non-root privileged admins.

- Accepting externally authenticated admins

Configure the security device to accept or not to accept root-privileged admins authenticated by a remote auth server.

- Default authentication

If remote authentication fails and local authentication is disabled, then device resorts to allowing the root-privileged admins to be authenticated locally. This fallback procedure is restricted to the serial console.

Defining Auth Server Objects

Before you can refer to external authentication (auth) servers in policies, IKE gateways, and L2TP tunnels, you must first define the auth server objects. The following examples illustrate how to define auth server objects for a RADIUS server, a SecurID server, an LDAP server, and a TACACS+ server.

Example: RADIUS Auth Server

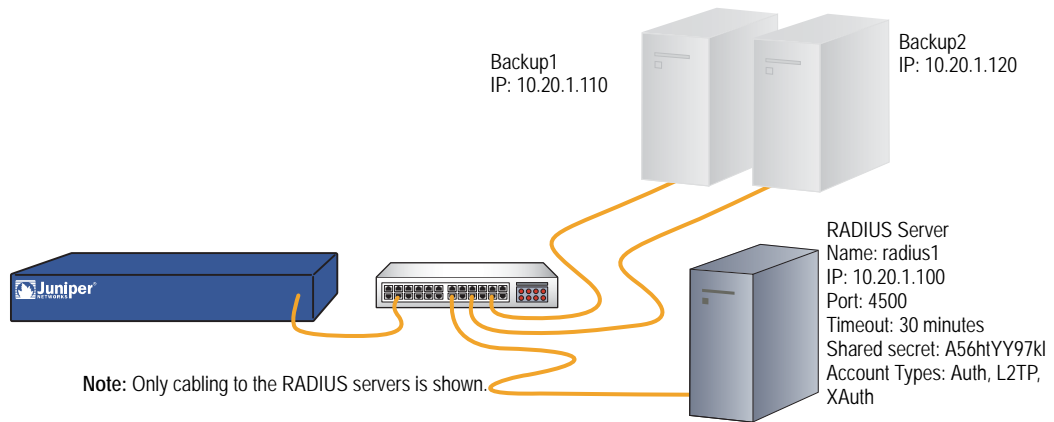
In the following example, you define an auth server object for a RADIUS server. You specify its user account types as auth, L2TP, and XAuth. You name the RADIUS server “radius1” and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 10.20.1.100, and change its port number from the default (1645) to 4500. You define its shared secret as “A56htYY97kl”. You also assign its two backup servers the IP addresses 10.20.1.110 and 10.20.1.120.

You change the authentication idle timeout value from the default (10 minutes) to 30 minutes and the RADIUS retry timeout from 3 seconds to 4 seconds. But because, with this setting, the session could theoretically remain open indefinitely (as long as one keystroke is sent every 30 minutes) you limit total session time by setting forced-timeout to 60 minutes. With this setting, after one hour the auth table entry for the user is removed, as are all associated sessions for the auth table entry, and the user needs to reauthenticate.

You also load the RADIUS dictionary file on the RADIUS server so that it can support queries for the following vendor-specific attributes (VSAs): user groups, admin privileges, and remote L2TP and XAuth settings.

In Figure 14, The security device sends auth, L2TP, and XAuth authentication requests to the primary RADIUS server, “radius1”, at 10.20.1.100. If the security device loses network connectivity with the primary RADIUS server, it redirects authentication requests to backup1 at 10.20.1.110. If the security device cannot reach backup1, it redirects authentication requests to backup2 at 10.20.1.120.

Figure 14: RADIUS Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: radius1
 IP/Domain Name: 10.20.1.100
 Backup1: 10.20.1.110
 Backup2: 10.20.1.120
 Timeout: 30
 Forced Timeout: 60
 Account Type: Auth, L2TP, XAuth
 RADIUS: (select)
 RADIUS Port: 4500
 Retry Timeout: 4 (seconds)
 Shared Secret: A56htYY97kl

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For more information, see “RADIUS Dictionary File” on page 21. For instructions on how to load the dictionary file onto a RADIUS server, refer to the documentation for your specific server.

CLI

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth l2tp xauth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 forced-timeout 60
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius timeout 4
set auth-server radius1 radius secret A56htYY97kl
save
```

NOTE: The order in which you enter the account types is important. For example, if you first enter **set auth-server radius1 account-type l2tp**, then your only subsequent choice is **xauth**; you cannot enter **auth** after **l2tp**. The correct order is easily remembered because it is alphabetical.

Changing the port number helps deter potential attacks targeted at the default RADIUS port number (1645).

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For more information, see “RADIUS Dictionary File” on page 21. For instructions on how to load the dictionary file onto a RADIUS server, refer to the documentation for your specific server.

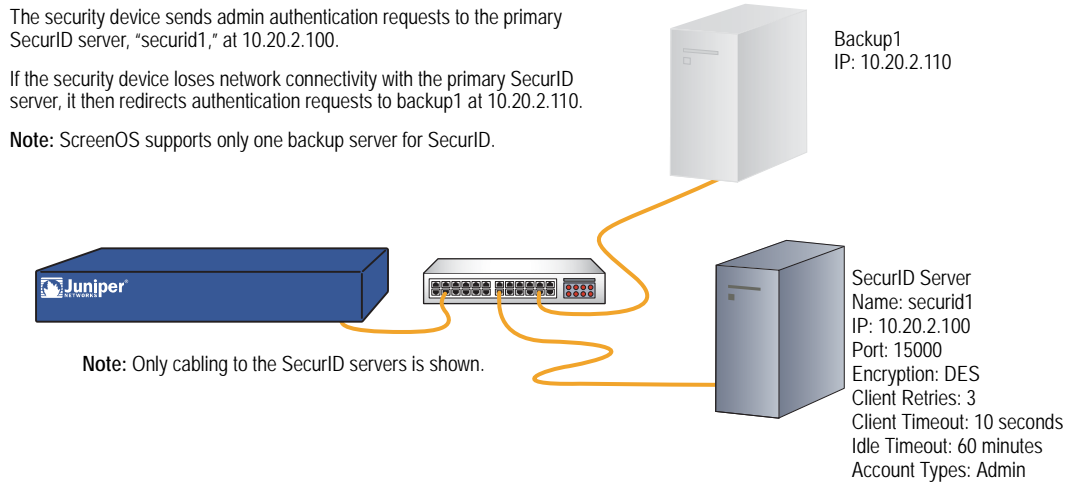
Example: SecurID Auth Server

In the following example, you configure an auth server object for a SecurID ACE server. You specify its user account type as admin. You name the server “securid1” and accept the ID number that the security device automatically assigns it. You enter the IP address of the primary server, which is 10.20.2.100, and the IP address of a backup server: 10.20.2.110. You change its port number from the default (5500) to 15000. The security device and the SecurID ACE server protect the authentication information using DES encryption. There are three allowable retries and a client timeout value of 10 seconds. You change the idle timeout value from the default (10 minutes) to 60 minutes. The **Use Duress** setting is disabled. See Figure 15.

NOTE: The client timeout value is the length of time in seconds that the SecurID client (that is, the security device) waits between authentication retry attempts.

The idle timeout value is the length of idle time in minutes that can elapse before the security device automatically terminates an inactive admin session. (For information comparing the timeout value as applied to admin users and other user types, see “Auth Server Object Properties” on page 18.)

Figure 15: SecurID Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: securid1
 IP/Domain Name: 10.20.2.100
 Backup1: 10.20.2.110
 Timeout: 60
 Account Type: Admin
 SecurID: (select)
 Client Retries: 3
 Client Timeout: 10 seconds
 Authentication Port: 15000
 Encryption Type: DES
 User Duress: No

CLI

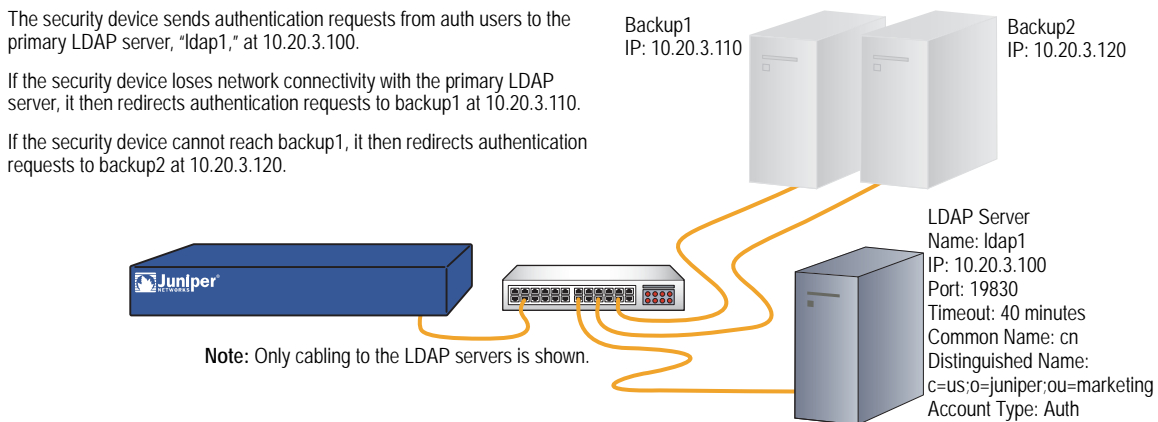
```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type admin
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
save
```

Example: LDAP Auth Server

In the following example, you configure an auth server object for an LDAP server. You specify its user account type as **auth**. You name the LDAP server "ldap1" and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 10.20.3.100, and change its port number from the default

(389) to 19830. You change the idle timeout value from the default (10 minutes) to 40 minutes. You also assign its two backup servers the IP addresses 10.20.3.110 and 10.20.3.120. The LDAP common name identifier is **cn**, and the Distinguished Name is **c=us;o=juniper;ou=marketing**. See Figure 16.

Figure 16: LDAP Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: ldap1
 IP/Domain Name: 10.20.3.100
 Backup1: 10.20.3.110
 Backup2: 10.20.3.120
 Timeout: 40
 Account Type: Auth
 LDAP: (select)
 LDAP Port: 4500
 Common Name Identifier: cn
 Distinguished Name (dn): c=us;o=juniper;ou=marketing

CLI

```
set auth-server ldap1 type ldap
set auth-server ldap1 account-type auth
set auth-server ldap1 server-name 10.20.3.100
set auth-server ldap1 backup1 10.20.3.110
set auth-server ldap1 backup2 10.20.3.120
set auth-server ldap1 timeout 40
set auth-server ldap1 ldap port 15000
set auth-server ldap1 ldap cn cn
set auth-server ldap1 ldap dn c=us;o=juniper;ou=marketing
save
```

Example: TACACS+ Auth Server

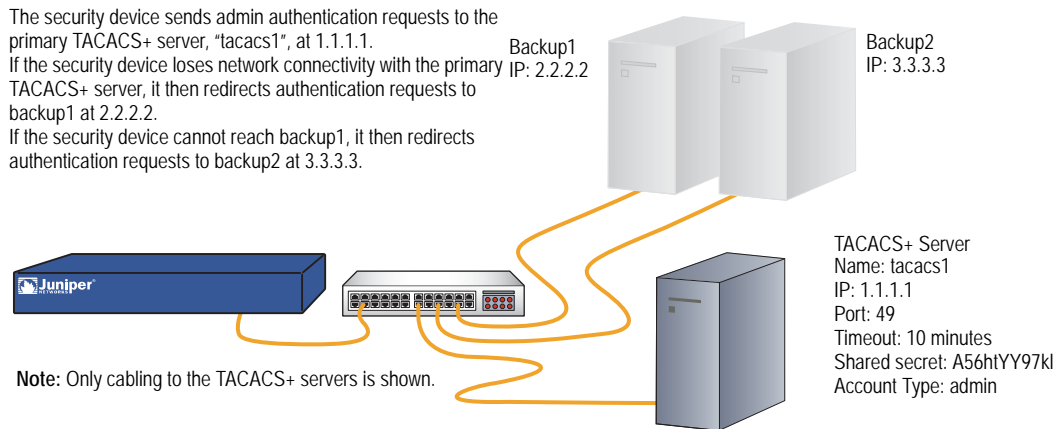
In the following example, you define an auth server object for a TACACS+ server. You name the TACACS+ server “tacacs1” and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 1.1.1.1. You define its shared secret as “A56htYY97kl”. You also assign its two backup servers the IP addresses 2.2.2.2 and 3.3.3.3.

You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your device are available.

You change the authentication idle timeout value from the default (10 minutes) to 30 minutes and the TACACS+ retry timeout from 3 seconds to 4 seconds. But because, with this setting, the session could theoretically remain open indefinitely (as long as one keystroke is sent every 30 minutes) you limit total session time by setting forced-timeout to 60 minutes. With this setting, after one hour the auth table entry for the user is removed, as are all associated sessions for the auth table entry, and the user needs to reauthenticate.

In Figure 17, The security device sends admin authentication requests to the primary TACACS+ server, “tacacs1”, at 1.1.1.1. If the security device loses network connectivity with the primary TACACS+ server, it redirects authentication requests to backup1 at 2.2.2.2. If the security device cannot reach backup1, it redirects authentication requests to backup2 at 3.3.3.3.

Figure 17: TACACS+ Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: tacacs1
 IP/Domain Name: 1.1.1.1
 Backup1: 2.2.2.2
 Backup2: 3.3.3.3
 Timeout: 10
 Forced Timeout: 60
 Account Type: admin
 TACACS: (select)
 TACACS Port: 49

Retry Timeout: 4 (seconds)
 Shared Secret: A56htYY97kl

CLI

```
set auth-server tacacs1 type tacacs
set auth-server tacacs1 account-type auth l2tp xauth
set auth-server tacacs1 server-name 1.1.1.1
set auth-server tacacs1 backup1 2.2.2.2
set auth-server tacacs1 backup2 3.3.3.3
set auth-server tacacs1 forced-timeout 60
set auth-server tacacs1 tacacs port 49
set auth-server tacacs1 tacacs secret A56htYY97kl
set admin auth server tacacs1
save
```

NOTE: Changing the port number helps deter potential attacks targeted at the default TACACS port number (49).

Defining Default Auth Servers

By default, the local database is the default auth server for all user types. You can specify external auth servers to be the default auth servers for one or more of the following user types:

- Admin
- Auth
- L2TP
- XAuth
- 802.1x

Then, if you want to use the default auth server for a specific user type when configuring authentication in policies, L2TP tunnels, or IKE gateways, you do not have to specify an auth server in every configuration. The security device refers to the appropriate auth servers that you previously appointed to be the defaults.

Example: Changing Default Auth Servers

In this example, you use the RADIUS, SecurID, and LDAP auth server objects that you created in the previous examples:

- radius1 (“Example: RADIUS Auth Server” on page 33)
- securid1 (“Example: SecurID Auth Server” on page 35)
- ldap1 (“Example: LDAP Auth Server” on page 36)

You then assign the local database, radius1, securid1, and ldap1 as the default servers for the following user types:

- Local: Default auth server for XAuth users

- radius1: Default auth server for L2TP users
- securid1: Default auth server for admin users
- ldap1: Default auth server for auth users

NOTE: By default, the local database is the default auth server for all user types. Therefore, unless you have previously assigned an external auth server as the default server for XAuth users, you do not need to configure it as such.

WebUI

VPNs > AutoKey Advanced > XAuth Settings: Select **Local** from the Default Authentication Server drop-down list, then click **Apply**.

VPNs > L2TP > Default Settings: Select **radius1** from the Default Authentication Server drop-down list, then click **Apply**.

Configuration > Admin > Administrators: Select **Local/securid1** from the Admin Auth Server drop-down list, then click **Apply**.

Configuration > Auth > Firewall: Select **ldap1** from the Default Auth Server drop-down list, then click **Apply**.

CLI

```
set xauth default auth server Local
set l2tp default auth server radius1
set admin auth server securid1
set auth default auth server ldap1
save
```

NOTE: By default, the local database is the default auth server for all user types. Therefore, unless you have previously assigned an external auth server as the default server for XAuth users, you do not need to configure it as such.

Chapter 3

Infranet Authentication

This chapter discusses how a Juniper Networks security device is deployed in a Unified Access Control (UAC) solution. The Juniper Networks UAC solution secures and ensures the delivery of applications and services across an enterprise infranet. UAC is an IP-based enterprise infrastructure that coordinates network, application, and endpoint intelligence and provides the control required to support network applications, manage network use, and reduce threats.

For more information about configuring and deploying UAC, refer to the *Unified Access Control Administration Guide*.

This chapter includes the following sections:

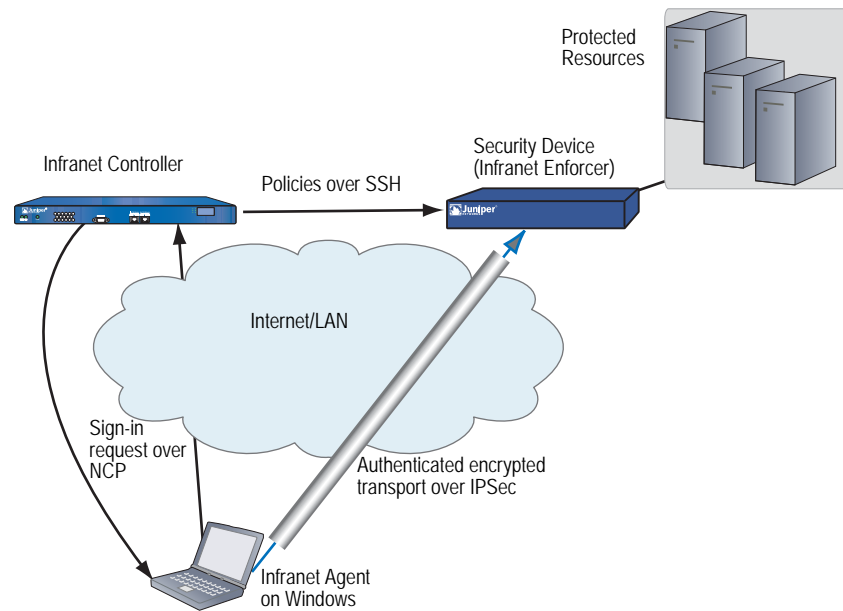
- “Unified Access Control Solution” on page 42
- “How the Security Device Works with the Infranet Controller” on page 43
- “Dynamic Discovery of Infranet Enforcers” on page 44
- “Infranet Controller Clustering” on page 45

Unified Access Control Solution

In a UAC deployment, your security device operates with the Infranet Controller to enforce policies. The security device, also called the *Infranet Enforcer*, is deployed in front of the servers and resources that you want to protect, as shown in Figure 18.

For more information about UAC and the Infranet Controller, see the *Unified Access Control Administration Guide*.

Figure 18: Deploying the Infranet Enforcer for Infranet Authentication



The security device and the Infranet Controller work together to provide granular endpoint security and firewall services to allow only qualified end users to access protected resources.

For example, you can configure the Infranet Controller with roles that define the level of access, such as full access to the protected resources for the *employee* role, and limited access for the *contractor* role. You can also create policies on the Infranet Controller that require endpoints to meet certain security requirements in order to have access to protected resources. For example, you can require an endpoint to use a minimum version of an antivirus (AV) application and have up-to-date AV definitions. If the endpoint does not meet the security requirements, you can display a page that instructs the user how to bring the endpoint into compliance.

The security device allows or denies access to resources based on role and endpoint security compliance. The security device connects with the Infranet Controller over an SSH connection that uses the NetScreen Address Change Notification (NACN) protocol. For information about configuring a connection between the two devices, refer to the *Unified Access Control Administration Guide*.

How the Security Device Works with the Infranet Controller

This section explains how the Juniper Networks security device and the Infranet Controller work together to establish communications and enforce security policies:

1. At startup, the Infranet Enforcer contacts the Infranet Controller over an SSL connection using the NACN protocol.

You can configure the Enforcer to work with up to eight Infranet Controllers. For more information about Infranet Controller failover, see “Infranet Controller Clustering on page 45

2. After the security device successfully establishes an NACN connection with the Controller, the Controller opens an SSH connection with the security device to push policy and user authentication information to the security device. For more information about how the Infranet Controller enables dynamic discovery of Enforcers, see “Dynamic Discovery of Infranet Enforcers on page 44
3. Whenever the Controller authenticates a user and verifies that the user's computer complies with endpoint security policies, the Controller pushes user-authentication information to the Enforcer. This information includes an auth table entry for each authenticated user.

An auth table entry consists of the user's role(s) and source IP address. It can store up to 65,536 distinct roles. The Internet Key Exchange (IKE) gateway, virtual private network (VPN), and tunnel infranet-auth policy enable the user's endpoint to protect network by establishing an Internet Protocol Security (IPSec) tunnel with the Enforcer as a way of protecting network traffic.

4. When the security device detects traffic from a user that matches an infranet-auth policy, it uses the user's auth table entry along with the access policies that apply to the protected resource to determine whether to allow the user to access the protected resource.

In addition to permitting and deny actions, the Controller policy allows you to bind additional action items, such as DI, IDP, AV, logging, Web filtering, and antispam to the Infranet Controller. Before you configure the Infranet Controller for these actions, you must configure a security policy for the associated action items and set up the device for infranet authentication.

The Infranet Controller policy can turn on or off specific features in the security policy. For example, if the AV action is defined for an Infranet Controller access policy (Resource policies > ScreenOS options), then the security device parses the user's traffic for viruses but does not perform DI, IDP, logging, Web filtering or antispam.

5. As necessary, the Infranet Controller sends commands to the security to remove policies or auth table entries and deny access to the protected resources. This can occur, for example, when the user's computer becomes non-compliant with endpoint security policies or loses its connection with the Infranet Controller, when you change the configuration of a user's role, or when you disable all user accounts on the Infranet Controller in response to a security problem such as virus on the network.

Dynamic Discovery of Infranet Enforcers

Dynamic discovery enables an Infranet Controller to dynamically provision access to endpoints through an Infranet Enforcer when the endpoint tries to access resource protected by the Enforcer. This saves the Controller from having to provision access through all Enforcers for all endpoints known to the Controller, which in turn saves space on Enforcers that would otherwise be used for unneeded auth table entries.

When an Infranet Controller connects to the Juniper Networks Infranet Enforcer (security device), the Controller sends commands to the Enforcer directing the Enforcer to notify the Controller when the Enforcer drops a packet.

To notify the Controller that the Enforcer has dropped a packet, the Enforcer sends a message to the Controller that contains the source IP address and interface of the endpoint from where the request originated, the destination IP address the endpoint was trying to access, the policy that caused the Enforcer to drop the packet, and an ID configured by the Controller which uniquely identifies the Enforcer to the Controller.

The Infranet Controller also configures the Infranet Enforcer to notify the Infranet Controller when auth table entries expire. The Controller configures idle timeout and expire notification settings in the Enforcer that remain in effect during the SSH connection with the Controller.

An auth table entry expires when there are no sessions in the Enforcer that match the auth table entry for the amount of time specified by the Controller in the idle timeout. Expiring auth table entries save space on the Enforcer that would otherwise be used for unneeded auth table entries. To notify the Controller that an auth table entry has expired, the Enforcer sends a message to the Controller containing the IP address of the expired auth table entry.

When the Infranet Enforcer receives a packet for a protected resource from an unauthenticated IP address, the Enforcer drops the packet and notifies the Controller. If the request is an HTTP request, the Infranet Enforcer may also send an HTTP response back to the endpoint redirecting the endpoint back to the URL you have configured, typically the login page of the Infranet Controller.

When the Infranet Controller receives notification that a packet has been dropped, the Controller configures an auth table entry in the Infranet Enforcer if the IP address of the endpoint is known to the Infranet Controller. Meanwhile, if the Enforcer has redirected the endpoint to the Controller's login page, the Infranet Controller also receives a request from the endpoint for the login page. When this happens, one of three possible scenarios takes place:

- The endpoint is not authenticated. The Infranet Controller prompts the endpoint for authentication credentials.

- The endpoint is authenticated, but no auth table entry is provisioned. The Infranet Controller provisions the auth table entry and redirects the endpoint back to the original URL requested by the endpoint.
- The endpoint is authenticated and already has an auth table entry. This happens because the Infranet Controller may have processed the dynamic discovery message before it received the HTTPs request from the endpoint. In this case, the Controller redirects the endpoint back to the original URL. In order for this to work, the Infranet Enforcer sends the Infranet Controller an ID that the Controller assigned to it. This ID is valid only as long as the Enforcer and the Controller are connected to each other.

For more information, refer to the *Unified Access Control Administration Guide*.

Infranet Controller Clustering

You can configure up to eight Infranet Controllers in a cluster to ensure connectivity between endpoints and protected resources is continuously maintained even if one of the Infranet Controllers fails. The security device communicates with only one Infranet Controller at a time; the other Infranet Controllers are used for failover within an Infranet Controller cluster. If the security device cannot connect to the first Infranet Controller, it tries the next one in its configuration list until a connection occurs. Infranet Controllers configured on the security device should all be members of the same Infranet Controller cluster.

Similar to the Infranet Controller clustering, an Infranet Controller cluster can communicate with a cluster of security devices that synchronize using the NetScreen Redundancy Protocol (NSRP) and keepalive messages between two types of clusters. The runtime objects (RTOs) such as auth table entries and infranet auth policies that the Infranet Controller sends to the security device are synchronized with all nodes in a security device cluster. A newly joined node establishes connection with an Infranet Controller only after it has synchronized the RTOs with its peers in the cluster to ensure that all peers in a security device cluster have the same state at the end of the synchronization. Any infranet auth table or auth policy updates from the Infranet Controller are then synchronized across all security devices in the cluster. For more information about RTO Synchronization, see *Volume 11: High Availability*.

Security devices in the cluster maintain the infranet auth table entries and infranet auth policies for two minutes after the keepalive timeout to allow for establishing a new connection to the Infranet Controller (or another node in the Infranet Controller cluster if the previously connected Infranet Controller has failed).

When an infranet auth table entry changes, all of its associated sessions are reevaluated. Any sessions that are no longer allowed are terminated.

Chapter 4

Authentication Users

An authentication user (or *auth user*) is a network user who must provide a username and password for authentication when initiating a connection across the firewall. You can store an auth user account on the local database or on an external RADIUS, SecurID, or LDAP server.

You can put several auth user accounts together to form an auth user group, which you can store on the local database or on a RADIUS server. A single auth user account can be in up to four user groups on the local database or on a RADIUS server. If you create an external user group on a RADIUS server, you must also create an identical—but unpopulated—user group on the security device. For example, if you define an auth user group named “au_grp1” on a RADIUS server named “rs1” and add 10 members to the group, then on the security device you need to define an auth user group also named “au_grp1,” identify it as an external user group, but add no members to it. When you reference the external auth user group “au_grp1” and auth server “rs1” in a policy, the security device can properly query the specified RADIUS server when traffic matching the policy provokes an authentication check. This chapter contains the following sections:

- “Referencing Auth Users in Policies” on page 48
 - “Run-Time Authentication” on page 48
 - “Pre-Policy Check Authentication (WebAuth)” on page 49
- “Referencing Auth User Groups in Policies” on page 50
 - “Example: Run-Time Authentication (Local User)” on page 51
 - “Example: Run-Time Authentication (Local User Group)” on page 52
 - “Example: Run-Time Authentication (External User)” on page 54
 - “Example: Run-Time Authentication (External User Group)” on page 55
 - “Example: Local Auth User in Multiple Groups” on page 58
 - “Example: WebAuth (Local User Group)” on page 60
 - “Example: WebAuth (External User Group)” on page 61
 - “Example: WebAuth + SSL Only (External User Group)” on page 63

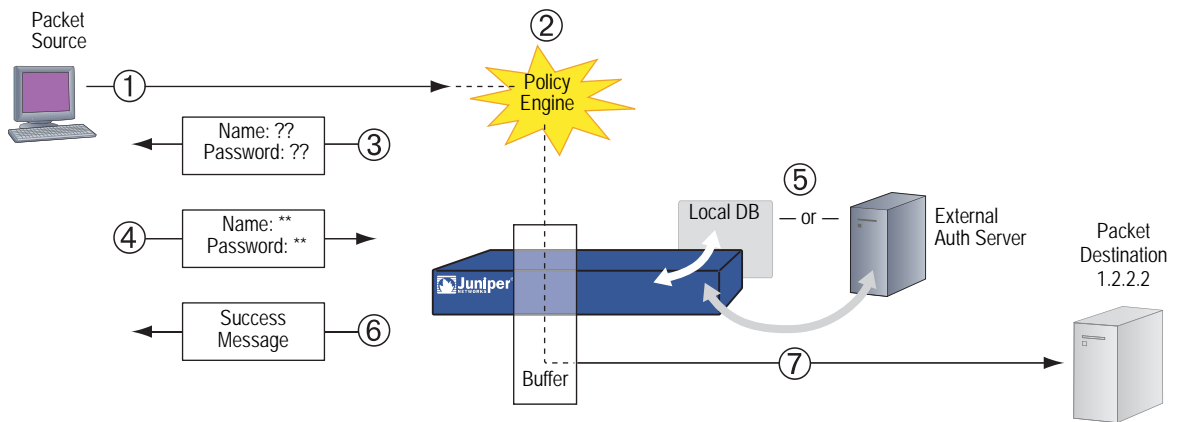
Referencing Auth Users in Policies

After you define an auth user, you can then create a policy that requires the user to authenticate himself or herself through one of two authentication schemes. The first scheme authenticates users when FTP, HTTP, or Telnet traffic matching a policy requiring authentication reaches the security device. In the second scheme, users authenticate themselves before sending traffic (of any kind—not just FTP, HTTP, or Telnet) to which a policy requiring user authentication applies.

Run-Time Authentication

When a user attempts to initiate an HTTP, an FTP, or a Telnet connection request to which a policy requiring authentication applies, the security device intercepts the request and prompts the user to enter a name and password (see “User Authentication” on page 2-170). Before granting permission, the security device validates the username and password by checking them against those stored in the local database or on an external auth server. See Figure 19.

Figure 19: Policy Lookup for a User

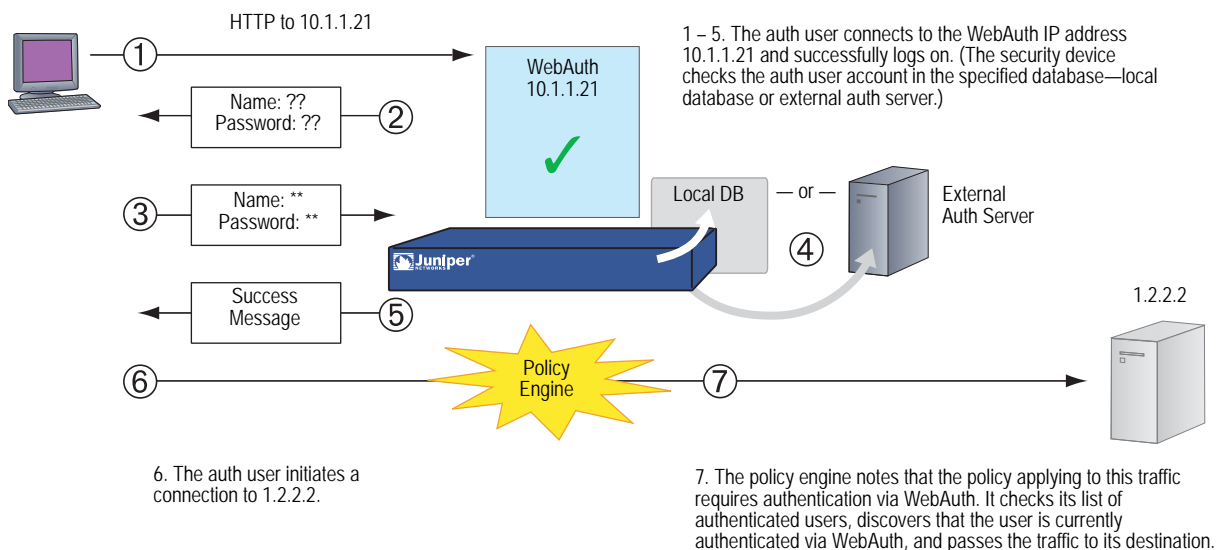


1. An auth user sends an FTP, an HTTP, or a Telnet packet to 1.2.2.2.
2. The security device intercepts the packet, notes that its policy requires authentication from either the local database or an external auth server, and buffers the packet.
3. The security device prompts the user for login information via FTP, HTTP, or Telnet.
4. The user replies with a username and password.
5. The security device either checks for an auth user account on its local database or it sends the login information to the external auth server as specified in the policy.
6. Finding a valid match (or receiving notice of such a match from the external auth server), the security device informs the user that the login has been successful.
7. The security device forwards the packet from its buffer to its destination of 1.2.2.2.

Pre-Policy Check Authentication (WebAuth)

Before sending traffic to an intended destination, an auth user initiates an HTTP session to the IP address hosting the WebAuth feature on the security device and authenticates himself or herself. After the security device authenticates the user, he or she can then send traffic to the destination as permitted by a policy requiring authentication via WebAuth. See Figure 20.

Figure 20: WebAuth Example



Some details about WebAuth:

- You can leave the default WebAuth auth server as the local database or you can choose an external auth server for the role. The main requirement for a WebAuth auth server is that the auth server must have auth user account-types.
- The WebAuth address must be in the same subnet as the interface that you want to use to host it. For example, if you want auth users to connect to WebAuth via ethernet3, which has IP address 1.1.1.1/24, then you can assign WebAuth an IP address in the 1.1.1.0/24 subnet.
- You can put a WebAuth address in the same subnet as the IP address of any physical interface, subinterface, or virtual security interface (VSI). (For information about different types of interfaces, see “Interfaces” on page 2-35.)
- If you want to use WebAuth while in Transparent mode, you can put a WebAuth address in the same subnet as the VLAN1 IP address.
- You can put WebAuth addresses on multiple interfaces.
- If you have multiple interfaces bound to the same security zone, you can put a WebAuth address in a subnet on one interface, and traffic from the same zone but using a different interface can still reach it.

- You should be aware that after a security device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication via WebAuth—from any other user at that same address. This might be the case if the users originate traffic from behind a NAT device that changes all original source addresses to a single translated address.
- You can direct the device to accept only SSL (HTTPS) connections for WebAuth sessions.

The security device redirects HTTP traffic to the WebAuth IP address (the WebAuth server) configured for the incoming interface. To set the security device to automatically redirect HTTP traffic to the WebAuth server:

WebUI

Policy > Policies > Edit > Advanced: Select the following options, then click **OK**:

- WebAuth (Local) (select)
- Redirect unauthenticated traffic (select)

CLI

```
set policy from zone1 to zone2 any any any permit webauth
redirect-unauthenticated
```

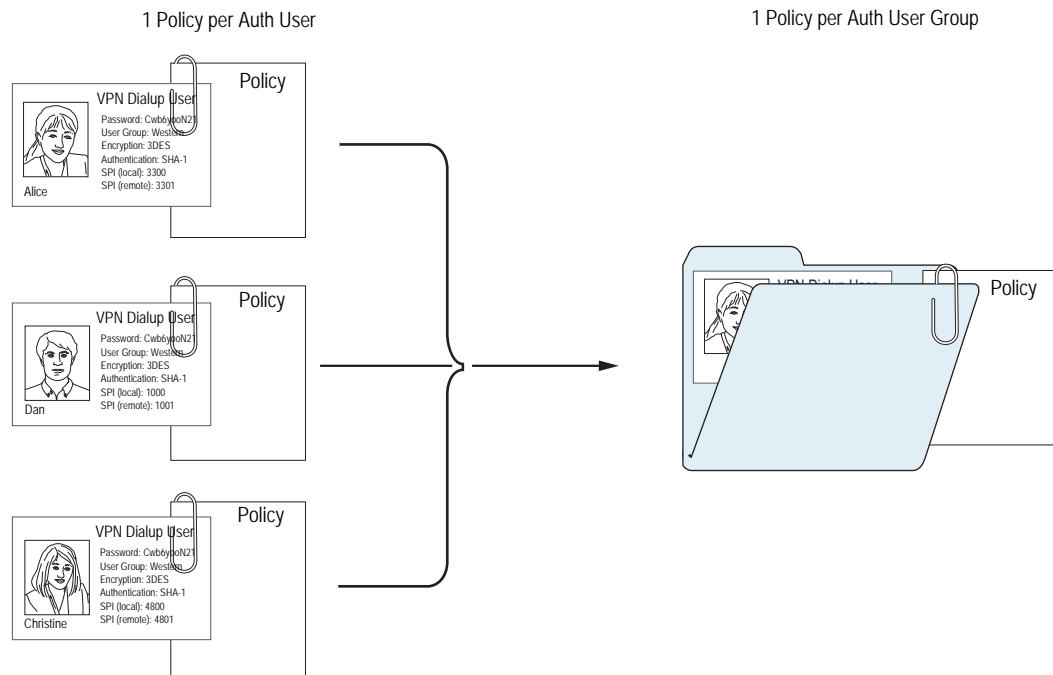
Referencing Auth User Groups in Policies

To manage a number of auth users, you can create auth user groups and store them either on the local security device or on an external RADIUS server.

NOTE: If you store users in groups on a RADIUS server, you must create unpopulated external user groups on the security device with names that correspond with those of the user groups you create on the RADIUS server.

Rather than manage each user individually, you can gather users into a group, so that any changes made to the group propagate to each group member. An auth user can be a member of up to four user groups on the local database or on a RADIUS server. An auth user who belongs to more than one group is required to supply a username and password only once, before being granted access to the resources defined for each group in which the user is a member. See Figure 21.

Figure 21: Auth User Groups



Example: Run-Time Authentication (Local User)

In this example, you define a local auth user named `louis` with password `iDa84rNk` and an address named `host1` in the Trust zone address book. You then configure two outgoing policies: one that denies all outbound traffic, and another from `host1` requiring `louis` to authenticate himself. (Louis must initiate all outbound traffic from `host1`.) The security device denies outbound access from any other address, as well as unauthenticated traffic from `host1`.

WebUI

1. Local Auth User and Address

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: `louis`
 Status: Enable
 Authentication User: (select)
 User Password: `iDa84rNk`
 Confirm Password: `iDa84rNk`

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: `host1`
 IP Address/Domain Name:
 IP/Netmask: (select), `10.1.1.4/32`
 Zone: Trust

2. Policies

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), Any
 Service: ANY
 Action: Deny

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), host1
 Destination Address:
 Address Book Entry: (select), Any
 Service: ANY
 Action: Permit
 Position at Top: (select)

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: Local
 User: (select), Local Auth User - louis

CLI

1. Local User and Address

```
set user louis password iDa84rNk
set address trust host1 10.1.1.4/32
```

NOTE: By default, a user to whom you assign a password is classified as an auth user.

2. Policies

```
set policy from trust to untrust any any any deny
set policy top from trust to untrust host1 any any permit auth user louis
save
```

Example: Run-Time Authentication (Local User Group)

In this example, you define a local user group named `auth_grp1`. You add previously created auth users `louis` and `lara` to the group. Then you configure a policy referencing `auth_grp1`. The policy provides FTP-GET and FTP-PUT privileges for `auth_grp1`, with address name “`auth_grp1`” (IP address 10.1.8.0/24) in the Trust zone to access an FTP server named “`ftp1`” (IP address 1.2.2.3/32) in the DMZ zone.

NOTE: When you create a user group in the local database, its user type remains undefined until you add a user to it. At that point, the user group takes the type or types of users that you add to it. You can create a multiple-type user group by adding auth, IKE, L2TP, and XAuth user types. You cannot combine Admin users with any other user type.

WebUI**1. Local User Group and Members**

Objects > Users > Local Groups > New: Enter **auth_grp1** in the Group Name field, do the following, then click **OK**:

Select **louis** and use the < < button to move him from the Available Members column to the Group Members column.

Select **lara** and use the < < button to move her from the Available Members column to the Group Members column.

2. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: auth_grp1
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.8.0/24
 Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ftp1
 IP Address/Domain Name:
 IP/Netmask: (select), 1.2.2.3/32
 Zone: DMZ

3. Policy

Policy > Policies > (From: Trust; To: DMZ) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), auth_grp1
 Destination Address:
 Address Book Entry: (select), ftp1
 Service: FTP
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: Local
 User Group: (select), Local Auth Group - auth_grp1

CLI**1. Local User Group and Members**

```
set user-group auth_grp1 location local
set user-group auth_grp1 user louis
set user-group auth_grp1 user lara
```

2. Address

```
set address trust auth_grp1 10.1.8.0/24
```

```
set address dmz ftp1 1.2.2.3/32
```

3. Policy

```
set policy top from trust to dmz auth_grp1 ftp1 ftp permit auth user-group
    auth_grp1
save
```

Example: Run-Time Authentication (External User)

In this example, you define an external LDAP auth server named “x_srv1” with the following attributes:

- Account type: auth
- IP address: 10.1.1.100
- Backup1 IP address: 10.1.1.110
- Backup2 IP address: 10.1.1.120
- Authentication timeout: 60 minutes
- LDAP port number: 14500
- Common name identifier: cn
- Distinguished name: c= us;o= netscreen

You load the auth user “euclid” with password eTcS114u on the external auth server. You then configure an outgoing policy that requires authentication on auth server x_srv1 for external user euclid.

WebUI

1. Auth Server

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

```
Name: x_srv1
IP/Domain Name: 10.1.1.100
Backup1: 10.1.1.110
Backup2: 10.1.1.120
Timeout: 60
Account Type: Auth
LDAP: (select)
    LDAP Port: 14500
    Common Name Identifier: cn
    Distinguished Name (dn): c=us;o=netscreen
```

2. External User

Define the auth user “euclid” with password eTcS114u on the external LDAP auth server x_serv1.

3. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: euc_host
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.1.20/32
 Zone: Trust

4. Policy

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), Any
 Service: ANY
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: x_srv1
 User: (select), External User
 External User: euclid

CLI

1. Auth Server

```
set auth-server x_srv1
set auth-server x_srv1 type ldap
set auth-server x_srv1 account-type auth
set auth-server x_srv1 server-name 10.1.1.100
set auth-server x_srv1 backup1 10.1.1.110
set auth-server x_srv1 backup2 10.1.1.120
set auth-server x_srv1 timeout 60
set auth-server x_srv1 ldap port 14500
set auth-server x_srv1 ldap cn cn
set auth-server x_srv1 ldap dn c=us;o=netscreen
```

2. External User

Define the auth user “euclid” with password eTcS114u on the external LDAP auth server x_serv1.

3. Address

```
set address trust euc_host 10.1.1.20/32
```

4. Policy

```
set policy top from trust to untrust euc_host any any auth server x_srv1 user euclid
save
```

Example: Run-Time Authentication (External User Group)

In this example, you configure an external RADIUS auth server named “radius1” and define an external auth user group named “auth_grp2.” You define the external auth user group auth_grp2 in two places:

1. External RADIUS auth server “radius1”
2. Security device

NOTE: The RADIUS auth server configuration is nearly identical to that in “Example: RADIUS Auth Server” on page 33, except that in this example you only specify “auth” as the user account type.

You populate the auth user group “auth_grp2” with auth users on the RADIUS server only, leaving the group unpopulated on the security device. The members in this group are accountants who require exclusive access to a server at IP address 10.1.1.80. You create an address book entry for the server and name the address “midas.” You then configure an intrazone policy permitting only authenticated traffic from auth_grp2 to midas, both of which are in the Trust zone. (For more information on intrazone policies, see “Policies” on page 2-159.)

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 21. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

If you are using a Microsoft IAS RADIUS server, there is no dictionary file to load. Instead, define the correct vendor-specific attributes (VSAs) on the server.

2. After you define auth user accounts on the RADIUS server, use the ScreenOS user group VSA to create the user group “auth_grp2” and apply it to the auth user accounts that you want to add to that group.

WebUI

1. Auth Server

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

```
Name: radius1
IP/Domain Name: 10.20.1.100
Backup1: 10.20.1.110
Backup2: 10.20.1.120
Timeout: 30
Account Type: Auth
RADIUS: (select)
    RADIUS Port: 4500
    Shared Secret: A56htYY97kl
```

2. External User Group

Objects > Users > External Groups > New: Enter the following, then click **OK**:

```
Group Name: auth_grp2
Group Type: Auth
```

3. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: midas
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.1.80/32
 Zone: Trust

4. Policy

Policy > Policies > (From: Trust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), midas
 Service: ANY
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: radius1
 User Group: (select), External Auth Group - auth_grp2

CLI**1. Auth-Server**

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

2. External User Group

```
set user-group auth_grp2 location external
set user-group auth_grp2 type auth
```

3. Address

```
set address trust midas 10.1.1.80/32
```

4. Policy

```
set policy top from trust to trust any midas any permit auth server radius1
user-group auth_grp2
save
```

Example: Local Auth User in Multiple Groups

In this example, you define a local auth user named Mary. Mary is a sales manager who needs access to two servers: server A, which is for the salespeople (sales_reps group), and server B, which is for the managers (sales_mgrs group). To provide access to both, you add Mary to the two user groups. You then create two policies, one for each group.

NOTE: This example does not show the configuration for the other group members.

WebUI

1. Local User

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: mary
 Status: Enable
 Authentication User: (select)
 User Password: iFa8rBd
 Confirm Password: iFa8rBd

2. Local User Groups and Member

Objects > Users > Local Groups > New: Enter **sales_mgrs** in the Group Name field, do the following, then click **OK**:

Select **mary** and use the << button to move her from the Available Members column to the Group Members column.

Objects > Users > Local Groups > New: Enter **sales_reps** in the Group Name field, do the following, then click **OK**:

Select **mary** and use the << button to move her from the Available Members column to the Group Members column.

3. Addresses

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: sales
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.8.0/24
 Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: server_a
 IP Address/Domain Name:
 IP/Netmask: (select), 1.1.1.5/32
 Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: server_b
 IP Address/Domain Name:

IP/Netmask: (select), 1.1.1.6/32
 Zone: Untrust

4. Policies

Policy > Policies > (From: Trust; To: Untrust) > New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), sales
 Destination Address:
 Address Book Entry: (select), server_a
 Service: FTP
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: Local
 User Group: (select), Local Auth Group - sales_reps

Policy > Policies > (From: Trust; To: Untrust) > New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), sales
 Destination Address:
 Address Book Entry: (select), server_b
 Service: FTP
 Action: Permit
 Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 Auth Server: (select)
 Use: Local
 User Group: (select), Local Auth Group - sales_mgrs

CLI

1. Local User

```
set user mary password iFa8rBd
```

2. Local User Groups and Member

```
set user-group sales_mgrs location local
set user-group sales_mgrs user mary
set user-group sales_reps location local
set user-group sales_reps user mary
```

3. Addresses

```
set address trust sales 10.1.8.0/24
set address untrust server_a 1.1.1.5/32
set address untrust server_b 1.1.1.6/32
```

4. Policy

```

set policy top from trust to untrust sales server_a ftp permit auth user-group
sales_reps
set policy top from trust to untrust sales server_b ftp permit auth user-group
sales_mgrs
save

```

Example: WebAuth (Local User Group)

In this example, you require users to preauthenticate themselves via the WebAuth method before initiating outbound traffic to the Internet. You create a user group named “auth_grp3” in the local database on the security device. You then create auth user accounts for everyone in the Trust zone and add them to auth_grp3.

The Trust zone interface uses ethernet1 and has IP address 10.1.1.1/24. You assign 10.1.1.50 as the WebAuth IP address, and you use keep the local database as the default WebAuth server. Consequently, before a user can initiate traffic to the Internet, he or she must first make an HTTP connection to 10.1.1.50 and log in with a username and password. The security device then checks the username and password against those in its database and either approves or rejects the authentication request. If it approves the request, the authenticated user has 30 minutes to initiate traffic to the Internet. After terminating that initial session, the user has another 30 minutes to initiate another session before the security device requires him or her to reauthenticate himself or herself.

WebUI**1. WebAuth**

Configuration > Auth > WebAuth: Select **Local** from the WebAuth Server drop-down list, then click **Apply**.

Network > Interfaces > Edit (for ethernet1): Select **WebAuth**, and in the WebAuth IP field enter **10.1.1.50**.

Configuration > Auth > Auth Servers > Edit (for Local): Enter **30** in the Timeout field, then click **Apply**.

2. User Group

Objects > Users > Local Groups > New: Enter **auth_grp3** in the Group Name field, do the following, then click **OK**:

Select **user name** and use the < < button to move that user from the Available Members column to the Group Members column.

Repeat the selection process, adding auth users until the group is complete.

3. Policy

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:

Address Book Entry: (select), Any

Destination Address:

Address Book Entry: (select), Any

Service: ANY
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
WebAuth: (select)
User Group: (select), Local Auth Group - auth_grp3

CLI

1. WebAuth

```
set webauth server Local
set interface ethernet1 webauth-ip 10.1.1.50
set interface ethernet1 webauth
set auth-server Local timeout 30
```

2. User Group

```
set user-group auth_grp3 location local
```

NOTE: The security device determines a local user group type by the type of members that you add to it. To make auth_grp3 an auth user group, add an auth user to the group.

Use the following command to add auth users to the user group you have just created:

```
set user-group auth_grp3 user name_str
```

3. Policy

```
set policy top from trust to untrust any any any permit webauth user-group
auth_grp3
save
```

Example: WebAuth (External User Group)

WebAuth is a method for pre-authenticating users before they initiate traffic across the firewall. In this example, you create a policy requiring authentication via the WebAuth method for all outgoing traffic.

You create an auth user group named “auth_grp4” on both the RADIUS server “radius1” and on the security device. On the RADIUS server, you create user accounts for everyone in the Trust zone and add them to auth_grp4.

NOTE: Nearly the same RADIUS server settings are used here as in “Example: RADIUS Auth Server” on page 33, except that in this example you only specify “auth” as the user account type.

The Trust zone interface uses ethernet1 and has IP address 10.1.1.1/24. You assign 10.1.1.50 as the WebAuth IP address, and you use the external RADIUS auth-server “radius1” as the default WebAuth server. Consequently, before a user can initiate traffic to the Internet, he or she must first make an HTTP connection to 10.1.1.50 and log in with a username and password. The security device then relays all WebAuth user authentication requests and responses between “radius1” and the users attempting to log in.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 21. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter user group “auth_grp4” on the auth-server “radius1”, and then populate it with auth user accounts.

WebUI

1. Auth-Server

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: radius1
 IP/Domain Name: 10.20.1.100
 Backup1: 10.20.1.110
 Backup2: 10.20.1.120
 Timeout: 30
 Account Type: Auth
 RADIUS: (select)
 RADIUS Port: 4500
 Shared Secret: A56htYY97k

2. WebAuth

Configuration > Auth > WebAuth: Select **radius1** from the WebAuth Server drop-down list, then click **Apply**.

Network > Interfaces > Edit (for ethernet1): Select **WebAuth**, in the WebAuth IP field enter **10.10.1.50**, then click **OK**.

3. User Group

Objects > Users > External Groups > New: Enter the following, then click **OK**:

Group Name: auth_grp4
 Group Type: Auth

4. Policy

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:

Address Book Entry: (select), Any
 Service: ANY
 Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 WebAuth: (select)
 User Group: (select), External Auth Group - auth_grp4

CLI

1. Auth-Server

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

2. WebAuth

```
set webauth server radius1
set interface ethernet1 webauth-ip 10.1.1.50
set interface ethernet1 webauth
```

3. User Group

```
set user-group auth_grp4 location external
set user-group auth_grp4 type auth
```

4. Policy

```
set policy top from trust to untrust any any any permit webauth user-group
  auth_grp4
save
```

Example: WebAuth + SSL Only (External User Group)

In this example, you combine WebAuth with Secure Sockets Layer (SSL) technologies to provide security for the usernames and passwords that users transmit when logging in. WebAuth makes use of the same certificate that secures administrative traffic to the security device for management via the WebUI. (For more information about SSL, see “Secure Sockets Layer” on page 3-5.)

The configuration for WebAuth using an external auth server plus SSL involves the following steps:

- You define an external RADIUS auth-server “radius1” and create an auth user group named “auth_grp5” on both the RADIUS server and on the security device. On the RADIUS server, you create user accounts for all auth users in the Untrust zone and add them to auth_grp5.

NOTE: Nearly identical RADIUS server settings are used here as in “Example: RADIUS Auth Server” on page 33, except that you only specify “auth” as the user account type here.

- The Untrust zone interface uses ethernet3 and has IP address 1.1.1.1/24. You assign 1.1.1.50 as the WebAuth IP address, instruct the device to accept only SSL connections for WebAuth authentication requests, and define the external RADIUS auth-server “radius1” as the default WebAuth server.
- You specify the following SSL settings:
 - IDX number (1 in this example) of a certificate that you have previously loaded on the security device
 - DES_SHA-1 ciphers
 - SSL port number 2020
- You then configure an incoming policy requiring authentication via the WebAuth + SSL method for all traffic from the Untrust to Trust zones.

NOTE: For information on how to obtain and load digital certificates onto a security device, see “Public Key Cryptography” on page 5-29.

Consequently, before a user can initiate traffic to the internal network, he or she must first make an HTTPS connection to https://1.1.1.50:2020 and log in with a username and password. The security device then relays all WebAuth user authentication requests and responses between “radius1” and the user attempting to log in.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the dictionary file, see “RADIUS Dictionary File” on page 21. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter user group “auth_grp5” on the auth-server “radius1,” and then populate it with auth user accounts.

WebUI

1. Auth-Server

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

```
Name: radius1
IP/Domain Name: 10.20.1.100
Backup1: 10.20.1.110
Backup2: 10.20.1.120
Timeout: 30
```

Account Type: Auth
 RADIUS: (select)
 RADIUS Port: 4500
 Shared Secret: A56htYY97k

2. WebAuth

Configuration > Auth > WebAuth: Select **radius1** from the WebAuth Server drop-down list, then click **Apply**.

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

WebAuth: (select)
 IP: 1.1.1.50
 SSL Only: (select)

3. SSL

Configuration > Admin > Management: Enter the following, then click **OK**:

HTTPS (SSL) Port: 2020
 Certificate: (select the certificate that you previously loaded)
 Cipher: DES_SHA-1

4. User Group

Objects > Users > External Groups > New: Enter the following, then click **OK**:

Group Name: auth_grp5
 Group Type: Auth

5. Policy

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Any
 Destination Address:
 Address Book Entry: (select), Any
 Service: ANY
 Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
 WebAuth: (select)
 User Group: (select), External Auth Group - auth_grp5

CLI

1. Auth-Server

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 21. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. WebAuth

```
set webauth server radius1
set interface ethernet3 webauth-ip 1.1.1.50
set interface ethernet3 webauth ssl-only
```

3. SSL

```
set ssl port 2020
set ssl cert 1
set ssl encrypt des sha-1
set ssl enable
```

4. User Group

```
set user-group auth_grp5 location external
set user-group auth_grp5 type auth
```

5. Policy

```
set policy top from untrust to trust any any any permit webauth user-group
    auth_grp5
save
```

Chapter 5

IKE, XAuth, and L2TP Users

This chapter covers the three types of users involved with tunneling protocols—Internet Key Exchange (IKE) users, XAuth users, and Layer 2 Transport Protocol (L2TP) users. It contains the following sections:

- “IKE Users and User Groups” on page 67
- “XAuth Users and User Groups” on page 70
 - “XAuth Users in IKE Negotiations” on page 72
 - “XAuth Client” on page 85
- “L2TP Users and User Groups” on page 86

NOTE: For more information and examples for IKE and L2TP, see *Volume 5: Virtual Private Networks*.

IKE Users and User Groups

An IKE user is a remote VPN user with a dynamically assigned IP address. The user—actually, the user’s device—authenticates itself by sending either a certificate or preshared key together with an IKE ID during Phase 1 negotiations with the security device.

The IKE ID can be an email address, an IP address, a domain name, or ASN1-DN string. A security device authenticates an IKE user if the user sends either of the following:

- A **certificate** in which one or more of the values that appear in the distinguished name (DN) fields or in the SubAltName field is the same as the user’s IKE ID configured on the security device
- A **preshared key** and an **IKE ID**, and the security device can successfully generate an identical preshared key from the received IKE ID and a preshared key seed value stored on the security device

NOTE: An example of an IKE ID using the Abstract Syntax Notation, version 1, distinguished name (ASN1-DN) format is
 CN= fiona,OU= it,O= juniper,L= sunnyvale,ST= ca,C= us,E= fiona@juniper.net.

You reference an IKE user or user group in an AutoKey IKE gateway configuration. By gathering IKE users that require similar gateway and tunnel configurations into a group, you only need to define one gateway referencing the group (and one VPN tunnel referencing that gateway), instead of one gateway and tunnel for each IKE user.

It is often impractical to create separate user accounts for every host. In such cases, you can create an IKE user group that has only one member, referred to as a group IKE ID user. The IKE ID of that user contains a set of values that must be present in the dialup IKE users' IKE ID definitions. If the IKE ID of a remote dialup IKE user matches the IKE ID of the group IKE ID user, ScreenOS authenticates that remote user. For more information, see "Group IKE ID" on page 5-193.

NOTE: You can only store IKE user and IKE user group accounts on the local database.

Example: Defining IKE Users

In this example, you define four IKE users, Amy, Basil, Clara, and Desmond, each with a different kind of IKE ID.

- Amy – email address (user-fully qualified domain name or U-FQDN): amy@juniper.net
- Basil – IP address: 3.3.1.1
- Clara – fully qualified domain name (FQDN): www.juniper.net
- Desmond – ASN1-DN string:
 CN= des,OU= art,O= juniper,L= sunnyvale,ST= ca,C= us,E= des@juniper.net

WebUI

Objects > Users > Local > New: Enter the following, then click **OK**:

```
User Name: Amy
Status: Enable
IKE User: (select)
Simple Identity: (select)
  IKE ID Type: AUTO
  IKE Identity : amy@juniper.net
```

Objects > Users > Local > New: Enter the following, then click **OK**:

```
User Name: Basil
Status: Enable
IKE User: (select)
Simple Identity: (select)
  IKE ID Type: AUTO
  IKE Identity: 3.3.1.1
```

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Clara
 Status: Enable
 IKE User: (select)
 Simple Identity: (select)
 IKE ID Type: AUTO
 IKE Identity: www.juniper.net

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Desmond
 Status: Enable
 IKE User: (select)
 Use Distinguished Name for ID: (select)
 CN: des
 OU: art
 Organization: juniper
 Location: sunnyvale
 State: ca
 Country: us
 E-mail: des@juniper.net

CLI

```
set user Amy ike-id u-fqdn amy@juniper.net
set user Basil ike-id ip 3.3.1.1
set user Clara ike-id fqdn www.juniper.net
set user Desmond ike-id wildcard
    CN=des,OU=art,O=juniper,L=sunnyvale,ST=ca,C=us,E=des@juniper.net
save
```

Example: Creating an IKE User Group

In this example, you create a user group named `ike_grp1`. It becomes an IKE user group when you add IKE user Amy to it. You then add the other three IKE uses that you defined in “Example: Defining IKE Users” on page 68.

WebUI

Objects > Users > Local Groups > New: Enter **ike_grp1** in the Group Name field, do the following, then click **OK**:

Select **Amy** and use the << button to move her from the Available Members column to the Group Members column.

Select **Basil** and use the << button to move him from the Available Members column to the Group Members column.

Select **Clara** and use the << button to move her from the Available Members column to the Group Members column.

Select **Desmond** and use the << button to move him from the Available Members column to the Group Members column.

CLI

```

set user-group ike_grp1 location local
set user-group ike_grp1 user amy
set user-group ike_grp1 user basil
set user-group ike_grp1 user clara
set user-group ike_grp1 user desmond
save

```

Referencing IKE Users in Gateways

After you define an IKE user or IKE user group, you can then reference it in an IKE gateway configuration when the remote IKE gateway is a dialup user or dialup user group.

To see examples that reference IKE users in gateway configurations, see:

- “Policy-Based Dialup VPN, AutoKey IKE” on page 5-170
- “Creating a Group IKE ID (Certificates)” on page 5-197
- “Group IKE ID with Certificates” on page 5-193

XAuth Users and User Groups

The XAuth protocol is composed of two components: remote VPN user authentication (username plus password) and TCP/IP address assignments (IP address, netmask, DNS server, and WINS server assignments). ScreenOS supports the application of either component by itself or both components in concert.

NOTE: The assigned netmask is always 255.255.255.255 and cannot be modified.

An XAuth user or user group is one or more remote users who authenticate themselves when connecting to the security device via an AutoKey IKE VPN tunnel and optionally receive TCP/IP settings from the security device. Whereas the authentication of IKE users is actually the authentication of VPN gateways or clients, the authentication of XAuth users is the authentication of the individuals themselves. XAuth users must enter information that only they are supposed to know—their username and password.

The ScreenOS-Remote client can use the TCP/IP settings it receives to create a virtual adapter when sending VPN traffic—while using the TCP/IP network adapter settings provided by the ISP or network admin for non-VPN traffic. By assigning known IP addresses to remote users, you can define routes on the security device to those addresses via specific tunnel interfaces. Then the security device can ensure that return routing reaches the remote user’s IP address through the VPN tunnel, not via the default gateway. Address assignments also allow a downstream firewall to reference those addresses when creating policies. You can control the length of time that an IP address is associated with an individual XAuth user with the XAuth lifetime setting.

NOTE: A virtual adapter is the TCP/IP settings (IP address, DNS server addresses, WINS server addresses) that the security device assigns to a remote user for the duration of a VPN tunnel connection. Only ScreenOS-Remote clients support virtual adapter functionality. Juniper Networks security platforms do not.

ScreenOS supports the following aspects of XAuth:

- Authentication of local XAuth users and external XAuth users
- Authentication of local XAuth user groups and external XAuth user groups if stored on a RADIUS auth server
- IP, DNS server, and WINS server address assignments from an IP address pool for local XAuth users and external XAuth users stored on a RADIUS auth server

To configure the security device to use default XAuth settings stored on an external RADIUS server, do either of the following:

- WebUI: On the VPNs > AutoKey Advanced > XAuth Settings page, select **Query Client Settings on Default Server**.
- CLI: Enter the **set xauth default auth server *name_str* query-config** command.

The security device can also use gateway-specific XAuth settings stored on an external RADIUS server. When configuring a specific IKE gateway, do either of the following:

- WebUI: On the VPNs > AutoKey Advanced > Gateway > New > Advanced page, select the name of the RADIUS server from the External Authentication drop-down list, and then select **Query Remote Setting**.
- CLI: Enter the **set ike gateway *name_str* xauth server *name_str* query-config** command.
- Authentication only without address assignments, address assignments only without authentication (**set ike gateway *name_str* xauth bypass-auth**), and both authentication and address assignments in combination

Event Logging for IKE Mode

When a remote user accesses the network through Internet Key Exchange (IKE), ScreenOS authenticates the user with XAuth; ScreenOS records the event details in the traffic log. The log details include the following:

- Gateway IP address
- Username
- Number of session retries
- Allocated client IP address from the local IP pool or RADIUS server

For more information about viewing the traffic log, see *Volume 3: Administration*.

XAuth Users in IKE Negotiations

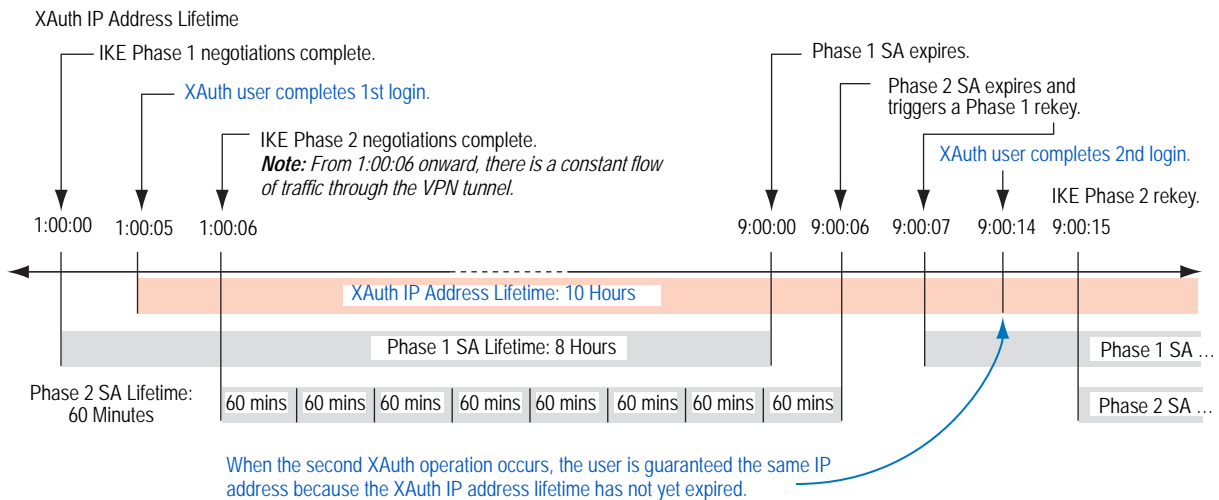
ScreenOS supports XAuth, version 6 (v6). To confirm that both parties in Phase 1 IKE negotiations support XAuth v6, they each send the following vendor ID to each other in the first two Phase 1 messages: 0x09002689DFD6B712. This vendor ID number is specified in the XAuth Internet draft, draft-beaulieu-ike-xauth-02.txt.

After the completion of Phase 1 negotiations, the security device sends a login prompt to the XAuth user at the remote site. If the XAuth user successfully logs on with the correct username and password, the security device assigns an IP address, 32-bit netmask, DNS server addresses, and WINS server addresses to the user, and the two parties continue with Phase 2 negotiations.

The XAuth user has 60 seconds to complete the login process. If the first login attempt fails, the XAuth user can make up to four more attempts, having 60 seconds for each attempt. If the user fails after five consecutive attempts, the security device stops providing a login prompt and severs the session.

At a minimum, the XAuth-assigned IP address belongs to a user for the duration specified as the XAuth address lifetime. The IP address might belong to the XAuth user longer, depending on when the Phase 1 and Phase 2 security associations (SAs) rekey. Figure 22 shows the relationship of Phase 1 and Phase 2 rekey operations and the XAuth IP address lifetime.

Figure 22: Phases 1 and 2 Rekey Operations and XAuth IP Address Lifetime



1. The Phase 1 SA is set with an 8-hour lifetime and expires after the first 8 hours.
2. The Phase 2 SA lifetime is set for 60 minutes. Because there is a 5-second delay during the initial IKE negotiations while the XAuth user enters his username and password, the eighth Phase 2 SA expires 8 hours and 6 seconds (5 seconds for the XAuth login + 1 second for Phase 2 negotiations) after Phase 1 negotiations complete.
3. Because there is active VPN traffic, the expiration of the eighth Phase 2 SA causes the Phase 1 SA, which expired 6 seconds prior, to rekey; that is, Phase 1 IKE negotiations (or “renegotiations”) occur.

4. After Phase 1 IKE renegotiations complete, the security device prompts the XAuth user to log in again.

NOTE: To avoid repeating further logins after the initial login, configure the VPN tunnel with any idletime other than 0 with the CLI command: **set vpn name gateway name idletime number (in minutes)**. If there is VPN activity at the completion of Phase 1 IKE renegotiations, the security device does not prompt the XAuth user to log in again. This option enables the user to download large files, transmit or receive streaming media, or participate in web conferences without interruption.

5. Because the XAuth address lifetime (10 hours) exceeds the Phase 1 SA lifetime, the user keeps the same IP address—although the user might get a different address after the next Phase 1 rekey occurs.

If the XAuth address lifetime had been shorter than the Phase 1 SA lifetime, the security device would have assigned the user another IP address, which might or might not have been the same as the previously assigned address.

NOTE: If it is crucial that a user always be assigned the same IP address, you can specify an address in the user configuration. The security device then assigns this address instead of assigning one at random from an IP pool. Note that such an address must not be in an IP pool or it might get assigned to another user and be unavailable when needed.

To change the address lifetime, do either of the following:

- (WebUI) VPNs > AutoKey Advanced > XAuth Settings: Enter a number (minutes) in the Reserve Private IP for XAuth User field, then click **Apply**.
- (CLI) `set xauth lifetime number`

To effectively disable the address lifetime feature, enter a value of 1—the minimum value allowed.

Example: XAuth Authentication (Local User)

In this example, you define an XAuth user named x1 with password aGgb80L0ws on the local database.

You then reference this user in a remote IKE gateway configuration to a peer at IP 2.2.2.2. You name the remote gateway **gw1**, specify Main mode and the proposal **pre-g2-3des-sha** for Phase 1 negotiations, and use the preshared key **juniper1**. You name the VPN tunnel **vpn1** and specify the Compatible set of proposals for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

WebUI

1. XAuth User

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: x1
 Status: Enable
 XAuth User: (select)
 User Password: iDa84rNk
 Confirm Password: iDa84rNk

2. VPN

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw1
 Security Level: Custom
 Remote Gateway Type:
 Static IP Address: (select), Address/Hostname: 2.2.2.2
 Preshared Key: juniper1
 Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom: (select)
 Phase 1 Proposal: pre-g2-3des-sha
 Mode (Initiator): Main (ID Protection)
 XAuth Server: (select)
 Local Authentication: (select)
 User: (select), x1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
 Security Level: Compatible
 Remote Gateway Tunnel: gw1

CLI

1. XAuth User

```
set user x1 password aGgb80LOws
set user x1 type xauth
unset user x1 type auth
```

NOTE: The CLI command **set user name_str password pswd_str** creates an auth user. To create an XAuth-only user, you must define the user as an XAuth user (**set user name_str type xauth**), and then remove the auth user definition (**unset user name_str type auth**).

2. VPN

```
set ike gate gw1 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper1
proposal pre-g2-3des-sha
set ike gateway gw1 xauth server Local user x1
set vpn vpn1 gateway gw1 sec-level compatible
save
```

Example: XAuth Authentication (Local User Group)

In this example, you create a user group named `xa-grp1` on the local database and add the XAuth user “`x1`” that you created in the previous example, “Example: XAuth Authentication (Local User)” on page 73. When you add that user to the group, it automatically becomes an XAuth user group.

You then reference this group in a remote IKE gateway configuration to a peer at IP `2.2.2.2`. You name the remote gateway “`gw2`,” specify Main mode and the proposal `pre-g2-3des-sha` for Phase 1 negotiations, and use the preshared key “`juniper2`.” You name the VPN tunnel “`vpn2`” and specify the “Compatible” set of proposals for Phase 2 negotiations. You choose the Untrust zone interface `ethernet3` as the outgoing interface.

WebUI

1. XAuth User Group

Objects > Users > Local Groups > New: Enter **`xa-grp1`** in the Group Name field, do the following, then click **OK**:

Select **`x1`** and use the < < button to move him from the Available Members column to the Group Members column.

2. VPN

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: `gw2`
 Security Level: Custom
 Remote Gateway Type:
 Static IP Address: (select), Address/Hostname: `2.2.2.2`
 Preshared Key: `juniper2`
 Outgoing Interface: `ethernet3`

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: `pre-g2-3des-sha`
 Mode (Initiator): Main (ID Protection)
 XAuth Server: (select)
 Local Authentication: (select)
 User Group: (select), `xa-grp1`

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: `vpn2`
 Security Level: Compatible
 Remote Gateway Tunnel:
 Predefined: (select), `gw2`

CLI**1. XAuth User Group**

```
set user-group xa-grp1 location local
set user-group xa-grp1 user x1
```

2. VPN

```
set ike gate gw2 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper2
proposal pre-g2-3des-sha
set ike gateway gw2 xauth server Local user-group xa-grp1
set vpn vpn2 gateway gw2 sec-level compatible
save
```

Example: XAuth Authentication (External User)

In this example, you reference an XAuth user named xa-1 with password iNWw10bd01 that you have previously loaded on an external SecurID auth server. This example uses almost the same configuration of the SecurID auth server as defined in “Example: SecurID Auth Server” on page 35, except that here you define the account type as XAuth.

You reference XAuth user xa-1 in a remote IKE gateway configuration to a peer at IP 2.2.2.2. You name the remote gateway **gw3**, specify Main mode and the proposal **pre-g2-3des-sha** for Phase 1 negotiations, and use the preshared key **juniper3**. You name the VPN tunnel **vpn3** and specify the proposal **g2-esp-3des-sha** for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

WebUI**1. External SecurID Auth Server**

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

```
Name: securid1
IP/Domain Name: 10.20.2.100
Backup1: 10.20.2.110
Timeout: 60
Account Type: XAuth
SecurID: (select)
  Client Retries: 3
  Client Timeout: 10 seconds
  Authentication Port: 15000
  Encryption Type: DES
  User Duress: No
```

2. XAuth User

Define the auth user “xa-1” with password iNWw10bd01 on the external SecurID auth server securid1.

3. VPN

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw3
 Security Level: Custom
 Remote Gateway Type:
 Static IP Address: (select), Address/Hostname: 2.2.2.2
 Preshared Key: juniper3
 Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: pre-g2-3des-sha
 Mode (Initiator): Main (ID Protection)
 XAuth Server: (select)
 External Authentication: (select), securid1
 User: (select)
 Name: xa-1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn3
 Security Level: Compatible
 Remote Gateway Tunnel:
 Predefined: (select), gw3

CLI**1. External SecurID Auth Server**

```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type xauth
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
```

2. XAuth User

Define the auth user “xa-1” with password iNWw10bd01 on the external SecurID auth-server securid1.

3. VPN

```
set ike gate gw3 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper3
proposal pre-g2-3des-sha
set ike gateway gw3 xauth server securid1 user xa-1
set vpn vpn3 gateway gw3 sec-level compatible
save
```

Example: XAuth Authentication (External User Group)

In this example, you configure an external RADIUS auth server named “radius1” and define an external auth user group named “xa-grp2.” You define the external XAuth user group xa-grp2 in two places:

1. External RADIUS auth server “radius1”
2. Security device

NOTE: The RADIUS auth server configuration is nearly identical to that in “Example: RADIUS Auth Server” on page 33, except that in this example you only specify “xauth” as the user account type.

You populate the XAuth user group “xa-grp2” with XAuth users on the RADIUS server only, leaving the group unpopulated on the security device. The members in this group are resellers at a remote site who require access to FTP servers in the corporate LAN. You add an entry in the Untrust zone address book for the remote site with IP address 10.2.2.0/24 and the name `reseller1`. You also enter an address in the Trust zone address book for the FTP server “rsl-srv1” with IP address 10.1.1.5/32.

You configure a VPN tunnel to 2.2.2.2 to authenticate XAuth users in the user group xa-grp2. You name the remote gateway “gw4,” specify Main mode and the proposal pre-g2-3des-sha for Phase 1 negotiations, and use the preshared key “juniper4.” You name the VPN tunnel “vpn4” and specify the “Compatible” set of proposals for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

Finally, you create a policy permitting FTP traffic from reseller1 in the Untrust zone via vpn4 to rsl-srv1 in the Trust zone.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 21. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter auth user group “xa-grp2” on the external auth server “radius1”, and then populate it with XAuth user accounts.

WebUI**1. Auth Server**

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: radius1
 IP/Domain Name: 10.20.1.100
 Backup1: 10.20.1.110
 Backup2: 10.20.1.120
 Timeout: 30
 Account Type: XAuth
 RADIUS: (select)
 RADIUS Port: 4500
 Shared Secret: A56htYY97kl

2. External User Group

Objects > Users > External Groups > New: Enter the following, then click **OK**:

Group Name: xa-grp2
 Group Type: XAuth

3. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: reseller1
 IP Address/Domain Name:
 IP/Netmask: (select), 10.2.2.0/24
 Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: rsl-svr1
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.1.5/32
 Zone: Trust

4. VPN

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw4
 Security Level: Custom
 Remote Gateway Type:
 Static IP Address: (select), Address/Hostname: 2.2.2.2
 Preshared Key: juniper4
 Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: pre-g2-3des-sha
 Mode (Initiator): Main (ID Protection)
 XAuth Server: (select)
 External Authentication: (select), securid1

User Group: (select)
Name: xa-grp2

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn4
Security Level: Compatible
Remote Gateway:
Predefined: (select), gw4

5. Policy

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), reseller1
Destination Address:
Address Book Entry: (select), rsl-svr1
Service: FTP-Get
Action: Tunnel
Tunnel VPN: vpn4
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. Auth Server

```
set auth-server radius1 type radius
set auth-server radius1 account-type xauth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

2. External User Group

```
set user-group xa-grp2 location external
set user-group xa-grp2 type xauth
```

3. Address

```
set address untrust reseller1 10.2.2.0/24
set address trust rsl-svr1 10.1.1.5/32
```

4. VPN

```
set ike gate gw4 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper4
proposal pre-g2-3des-sha
set ike gateway gw4 xauth server radius1 user-group xa-grp2
set vpn vpn4 gateway gw4 sec-level compatible
```

5. Policy

```
set policy top from untrust to trust reseller1 rsl-svr1 ftp-get tunnel vpn vpn4
save
```

Example: XAuth Authentication and Address Assignments (Local User Group)

In this example, you set up both authentication and IP, DNS server, and WINS server IP address assignments for an IKE/XAuth user group stored on the local database.

NOTE: You can also use an external RADIUS auth server for XAuth user authentication and address assignments. You can use an external SecurID or LDAP auth server for XAuth authentication only (not for address assignments). For IKE user authentication, you can only use the local database.

When an IKE/XAuth user makes a dialup VPN connection to the security device, the security device authenticates the IKE user (that is, the client device) using an IKE ID and an RSA certificate during Phase 1 negotiations. The security device then authenticates the XAuth user (that is, the individual using the device) using a username and password and assigns IP, DNS server, and WINS server IP addresses between Phase 1 and Phase 2 negotiations.

You create a local user group `ixa-grp1`. You then define two IKE/XAuth users named “`ixa-u1`” (password: `ccF1m84s`) and “`ixa-u2`” (password: `C113g1tw`) and add them to the group, thereby defining the group type as IKE/XAuth. (The addition of other IKE/XAuth users to the group is not included in the example.)

You create a DIP pool named `xa-pool1` with an address range from `10.2.2.1` to `10.2.2.100`. This is the pool of addresses from which the security device draws an IP address when assigning one to an XAuth user.

NOTE: The DIP pool must be in a different address space than that of the zone to which the XAuth user directs traffic to avoid routing problems and duplicate address assignments.

You configure the following XAuth default settings:

- Set the XAUTH address timeout to 480 minutes.
- Select the local database as the default auth server.
- Enable Challenge Handshake Authentication Protocol (CHAP), in which the security device sends a challenge (encryption key) to the remote client, who uses the key to encrypt his or her login name and password.
- Select `xa-pool1` as the default DIP pool.
- Define the primary and secondary DNS server IP addresses as `10.1.1.150` and `10.1.1.151`, respectively.
- Define the primary and secondary WINS server IP addresses as `10.1.1.160` and `10.1.1.161`, respectively.

You configure an IKE gateway named “ixa-gw1,” referencing user group ixa-grp1 and using the default XAuth auth server settings. You then configure a VPN tunnel name named “ixa-tun1” and a policy permitting traffic from ixa-grp1 to the Trust zone (IP address 10.1.1.0/24) via VPN tunnel ixa-tun1.

WebUI

1. IKE/XAuth Users and User Group

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: ixa-u1
 Status: Enable
 IKE User: (select)
 Simple Identity: (select)
 IKE ID Type: AUTO
 IKE Identity: u1@juniper.net
 XAuth User: (select)
 User Password: ccF1m84s
 Confirm Password: ccF1m84s

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: ixa-u2
 Status: Enable
 IKE User: (select)
 Simple Identity: (select)
 IKE ID Type: AUTO
 IKE Identity: u2@juniper.net
 XAuth User: (select)
 User Password: C113g1tw
 Confirm Password: C113g1tw

Objects > Users > Local Groups > New: Enter **ixa-grp1** in the Group Name field, do the following, then click **OK**:

Select **ixa-u1** and use the << button to move him from the Available Members column to the Group Members column.

Select **ixa-u2** and use the << button to move him from the Available Members column to the Group Members column.

2. IP Pool

Objects > IP Pools > New: Enter the following, then click **OK**:

IP Pool Name: xa-pool1
 Start IP: 10.2.2.1
 End IP: 10.2.2.100

3. Default XAuth Auth Server

VPNs > AutoKey Advanced > XAuth Settings: Enter the following, then click **Apply**:

Reserve Private IP for XAuth User: 480 Minutes
 Default Authentication Server: Local
 Query Client Settings on Default Server: (clear)
 CHAP: (select)
 IP Pool Name: xa-pool1
 DNS Primary Server IP: 10.1.1.150
 DNS Secondary Server IP: 10.1.1.151
 WINS Primary Server IP: 10.1.1.160
 WINS Secondary Server IP: 10.1.1.161

4. Address

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_zone
 IP Address/Domain Name:
 IP/Netmask: (select), 10.1.1.0/24
 Zone: Trust

5. VPN

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: ixa-gw1
 Security Level: Custom
 Remote Gateway Type:
 Dialup User Group: (select)
 Group: ixa-grp1

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: rsa-g2-3des-sha
 Mode (Initiator): Aggressive
 Outgoing Interface: ethernet3
 XAuth Server: (select)
 User Default: (select)
 User Group: (select), ixa-grp1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: ixa-vpn1
 Security Level: Compatible
 Remote Gateway:
 Predefined: (select), ixa-gw1

6. Policy

Policy > Policies > (From: Untrust; To: Trust) New: Enter the following, then click **OK**:

Source Address:
 Address Book Entry: (select), Dial-Up VPN
 Destination Address:
 Address Book Entry: (select), Trust_zone
 Service: ANY
 Action: Tunnel
 Tunnel VPN: ixa-vpn1
 Modify matching bidirectional VPN policy: (clear)
 Position at Top: (select)

CLI**1. IKE/XAuth Users and User Group**

```
set user-group ixa-grp1 location local
set user ixa-u1 type ike xauth
set user ixa-u1 ike-id u-fqdn u1@ns.com
set user ixa-u1 password ccF1m84s
unset user ixa-u1 type auth
set user ixa-u2 type ike xauth
set user ixa-u2 ike-id u-fqdn u2@juniper.net
set user ixa-u2 password C113g1tw
unset user ixa-u2 type auth
```

2. IP Pool

```
set ippool xa-pool1 10.2.2.1 10.2.2.100
```

3. Default XAuth Auth Server

```
set xauth lifetime 480
set xauth default auth server Local chap
set xauth default ippool xa-pool1
set xauth default dns1 10.1.1.150
set xauth default dns2 10.1.1.151
set xauth default wins1 10.1.1.160
set xauth default wins2 10.1.1.161
```

4. Address

```
set address trust Trust_zone 10.1.1.0/24
```

5. VPN

```
set ike gateway ixa-gw1 dialup ixa-grp1 aggressive outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway ixa-gw1 xauth server Local user-group ixa-grp1
set vpn ixa-vpn1 gateway ixa-gw1 sec-level compatible
```

6. Policy

```
set policy top from untrust to trust "Dial-Up VPN" Trust_zone any tunnel vpn
  ixa-vpn1
save
```

XAuth Client

An XAuth client is a remote user or device that connects to an XAuth server via an AutoKey IKE VPN tunnel. A security device can act as an XAuth client, responding to authentication requests from a remote XAuth server. After the completion of Phase 1 negotiations, the remote XAuth server sends a login prompt to the security device. If the security device acting as an XAuth client successfully logs in with the correct username and password, Phase 2 negotiations commence.

To configure the security device as an XAuth client, you must specify the following:

- IKE gateway name
- XAuth username and password

You can configure the following types of XAuth authentication:

- **Any** — Allows either Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP)
- **CHAP** — Allows CHAP only

Example: Security Device as an XAuth Client

In this example, you first configure a remote IKE gateway *gw1* with IP address 2.2.2.2. You specify the standard security level and use the preshared key *juniper1*. You then configure an XAuth client for the IKE gateway with the username *beluga9* and the password *1234567*. You also require CHAP authentication for the client.

WebUI

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw1
 Security Level: Standard (select)
 Remote Gateway Type:
 Static IP Address: (select), Address/Hostname: 2.2.2.2
 Preshared Key: juniper1
 Outgoing Interface: Untrust

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

XAuth Client: (select)
 User Name: beluga9
 Password: 1234567
 Allowed Authentication Type: (select), CHAP Only

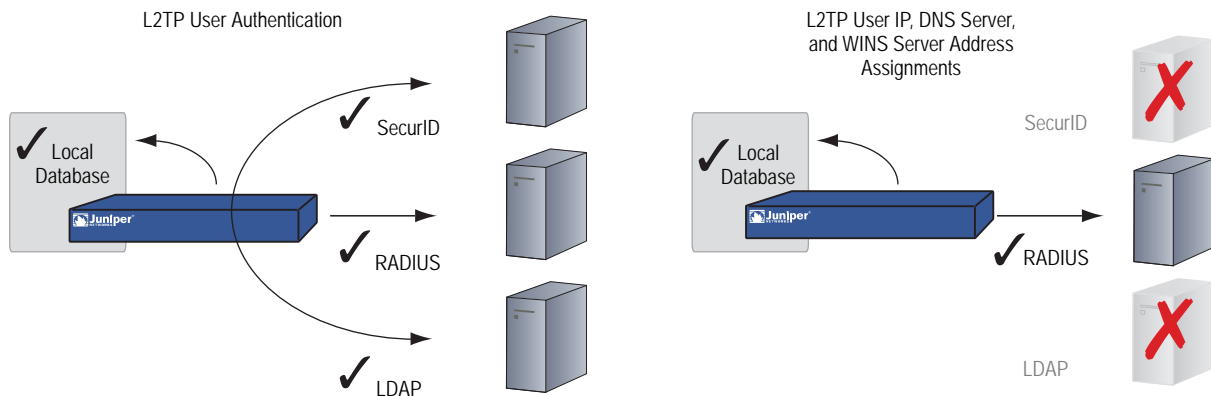
CLI

```
set ike gateway gw1 ip 2.2.2.2 Main outgoing-interface untrust preshare juniper1
sec-level standard
set ike gateway gw1 xauth client chap username beluga9 password 1234567
save
```

L2TP Users and User Groups

Layer 2 Tunneling Protocol (L2TP) provides a means for authenticating remote users and assigning IP, DNS server, and WINS server addresses. You can configure the security device to use either the local database or an external auth server to authenticate L2TP users. To make IP, DNS server, and WINS server address assignments, you can configure the security device to use either the local database or a RADIUS server (loaded with the RADIUS dictionary file—see “RADIUS Dictionary File” on page 21). See Figure 23.

Figure 23: Authenticating Users with L2TP



You can even use a combination of auth servers, a different one for each of the two aspects of L2TP. For example, you might use a SecurID server to authenticate an L2TP user but make the address assignments from the local database. The following example illustrates the application of two auth servers to handle both components of L2TP. For other examples, along with a detailed examination of L2TP, see “Layer 2 Tunneling Protocol” on page 5-215.

Example: Local and External L2TP Auth Servers

In this example, you set up an external SecurID auth server to authenticate L2TP users, and you use the local database to assign L2TP users with IP, DNS server, and WINS server addresses.

The external SecurID auth server is securid1. It is nearly identical to the auth server configuration in “Example: SecurID Auth Server” on page 35 except that the account type is L2TP. The SecurID auth server parameters are as follows:

- Name: securid1
- IP Address: 10.20.2.100
- Backup1 IP Address: 10.20.2.110
- Port: 15000
- Client Retries: 3
- Client Timeout: 10 seconds

- Idle Timeout: 60 minutes
- Account Type: L2TP
- Encryption: DES

The L2TP default settings are as follows:

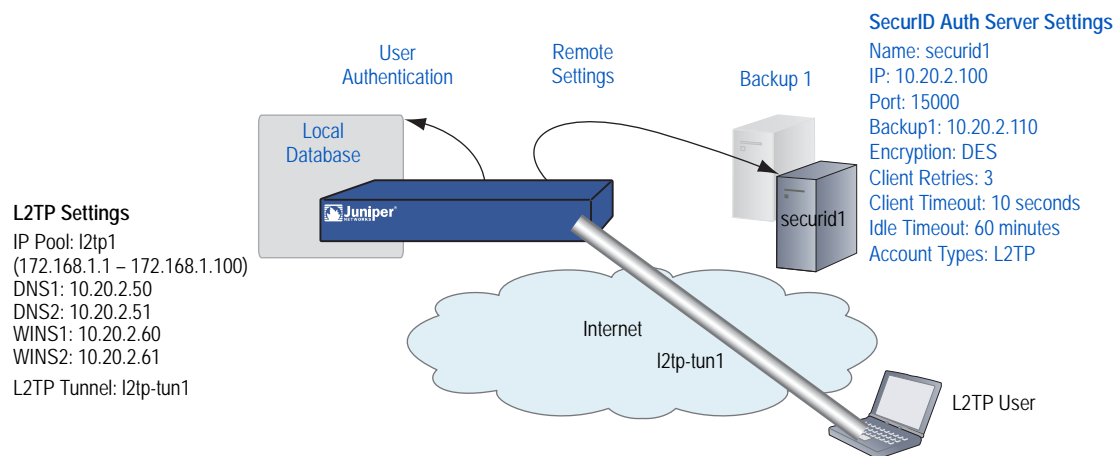
- IP Pool: I2tp1 (172.168.1.1 – 172.168.1.100)
- DNS Primary Server IP: 10.20.2.50
- DNS Secondary Server IP: 10.20.2.51
- PPP Authentication: CHAP
- WINS Primary Server IP: 10.20.2.60
- WINS Secondary Server IP: 10.20.2.61

After configuring the security device with the above settings, you create an L2TP tunnel named “l2tp-tun1” that references securid1 for authentication and uses the default settings for address assignments.

You must also set up the SecurID server as shown above and populate it with L2TP users. Figure 24 shows the L2TP settings, SecurID Auth server settings, and network setup.

NOTE: An L2TP-only configuration is not secure. To add security to an L2TP tunnel, we recommend that you combine it with an IPSec tunnel, which must be in Transport mode, as shown in “Configuring L2TP-over-IPSec” on page 5-228.

Figure 24: Local and External L2TP Servers



WebUI

1. Auth Server

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: securid1
 IP/Domain Name: 10.20.2.100
 Backup1: 10.20.2.110
 Timeout: 60
 Account Type: L2TP
 SecurID: (select)
 Client Retries: 3
 Client Timeout: 10 seconds
 Authentication Port: 15000
 Encryption Type: DES
 Use Duress: No

2. IP Pool

Objects > IP Pools > New: Enter the following, then click **OK**:

IP Pool Name: l2tp1
 Start IP: 172.168.1.1
 End IP: 172.168.1.100

3. L2TP Default Settings

VPNs > L2TP > Default Settings: Enter the following, then click **Apply**:

Default Authentication Server: Local
 IP Pool Name: l2tp1
 PPP Authentication: CHAP
 DNS Primary Server IP: 10.20.2.50
 DNS Secondary Server IP: 10.20.2.51
 WINS Primary Server IP: 10.20.2.60
 WINS Secondary Server IP: 10.20.2.61

4. L2TP Tunnel

VPNs > L2TP > Tunnel > New: Enter the following, then click **OK**:

Name: l2tp-tun1
 Use Custom Settings: (select)
 Authentication Server: securid1
 Query Remote Settings: (clear)
 Dialup User: (select), Allow Any

CLI

1. Auth Server

```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type l2tp
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
```

2. IP Pool

```
set ippool l2tp1 172.168.1.1 172.168.1.100
```

3. L2TP Default Settings

```
set l2tp default auth server Local  
set l2tp default ippool l2tp1  
set l2tp default ppp-auth chap  
set l2tp dns1 10.20.2.50  
set l2tp dns1 10.20.2.51  
set l2tp wins1 10.20.2.60  
set l2tp wins2 10.20.2.61
```

4. L2TP Tunnel

```
set l2tp l2tp-tun1  
set l2tp l2tp-tun1 auth server securid1  
save
```


Chapter 6

Extensible Authentication for Wireless and Ethernet Interfaces

This chapter explains the options available for and examples of using Extensible Authentication Protocol (EAP) to provide authentication for Ethernet and wireless interfaces. It contains the following sections:

- “Overview” on page 92
- “Supported EAP Types” on page 92
- “Enabling and Disabling 802.1X Authentication” on page 93
- “Configuring 802.1X Settings” on page 94
- “Configuring Authentication Server Options” on page 98
- “Viewing 802.1X Information” on page 100
- “Configuration Examples” on page 102
 - “Configuring the Security Device with a Directly Connected Client and RADIUS Server” on page 102
 - “Configuring a Security Device with a Hub Between a Client and the Security Device” on page 103
 - “Configuring the Authentication Server with a Wireless Interface” on page 104

Overview

EAP is an authentication framework that supports multiple authentication methods. EAP typically runs directly over data link layers, such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring Layer 3 addressing.

IEEE 802.1X works for port-based access control, and IKEv2 uses it as an option for authentication. EAP functions in a security device configured in Transparent or Route (with or without Network Address Translation enabled) mode. NetScreen Redundancy Protocol (NSRP) supports EAP in networks with high availability. Log messages and SNMP support are also available.

802.1X support is available for all platforms.

EAP functions as the authentication portion of PPP, which operates at Layer 2. EAP authenticates a supplicant, or client, after the supplicant sends proper credentials and the authentication server, usually a RADIUS server, defines the user-level permissions. When you use EAP, all authentication information passes through the security device (known as a pass-through method of EAP authentication). All user information is stored on the authentication server.

If you use a RADIUS server for authentication that supports vendor-specific attributes (VSAs), you can use the zone-verification feature, which verifies the zones a client is a member of.

Supported EAP Types

The EAP types described in Table 11 are supported.

Table 11: EAP Types

Type	Description
EAP-TLS (Transport Layer Security)	The most common EAP derivative and is supported by most RADIUS servers. EAP-TLS uses certificates for user and server authentication and for dynamic session key generation.
EAP-TTLS (Tunneled Transport Layer Security)	Requires only a server-side certificate and a valid username and password for authentication. Steel-Belted RADIUS supports TTLS.
EAP-PEAP (Protected EAP)	Designed to compensate for the lack of features in EAP-TLS and reduce management complexity. It requires only server-side certificates and a valid username and password. It provides support for key exchange, session resumption, fragmentation, and reassembly. Steel-Belted RADIUS and Microsoft IAS support Protected EAP.
EAP-MD5 (Message Digest Algorithm 5)	Algorithm that uses a challenge and response process to verify MD5 hashes.

Enabling and Disabling 802.1X Authentication

By default, 802.X authentication is disabled. You can enable 802.1X authentication for Ethernet and wireless interfaces using the WebUI or CLI. To enable 802.1X authentication on an Ethernet interface, you modify the interface's configuration. 802.1X on a wireless interface is automatically enabled after you create and configure an SSID and then bind it to the wireless interface.

Ethernet Interfaces

Use one of the following procedures to enable 802.1X on the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (ethernet1) > 802.1X: Click **Enable**.

CLI

To enable 802.1X on the ethernet1 interface, enter the following command:

```
set interface ethernet1 dot1x
```

To disable 802.1X on the ethernet1 interface, enter the following command:

```
unset interface ethernet1 dot1x
```

Wireless Interfaces

Use one of the following procedures to enable 802.1X on a wireless interface. The following procedures create an SSID named **hr**, using WPA as the authentication method and TKIP as the encryption method. The authentication server is a predefined RADIUS server named **radius1**. The SSID is then bound to wireless interface 0/1.

WebUI

Wireless > SSID > Click **New**. Enter the following, then click **OK**.

```
SSID Name: hr
WPA Based Authentication and Encryption Methods: Select WPA, TKIP
Auth Server: Select radius1.
Wireless Interface Binding: Select wireless0/1.
```

Depending on the security device, you need to activate the changes you configured by clicking the Activate Changes button at the top of the page or by selecting Wireless > Activate Changes.

To disable 802.1X on a wireless interface, select **none** from the Wireless Interface Binding list.

CLI

To enable 802.1X on a wireless interface, you must create and configure an SSID and then bind it to the wireless interface.

```
set ssid name hr
set ssid hr authentication wpa encryption tkip auth-server radius1
set ssid hr interface wireless0/1
```

To disable 802.1X on the wireless interface, enter one of the following command, which unbinds the SSID to the interface:

```
unset ssid hr interface
```

For security devices with two radio transceivers, you can also enter the following command, which unbinds the interface to the radio:

```
unset interface wireless0/1 wlan
```

You can also disable 802.1X on a wireless interface by changing the authentication method of an SSID that does not require use of an authentication server. Disabling the wireless interface with the **set interface wireless_interface shutdown** command also disables 802.1X.

Configuring 802.1X Settings

For Ethernet interfaces, you can optionally configure 802.1X settings. For wireless interfaces, these settings cannot be modified, and the default values are used, as listed in Table 12.

Table 12: 802.1X Settings

Option	Default Value	Alternative Values
Port control	auto	force-unauthorized
Control mode	virtual	interface
Maximum user	16 (Ethernet); 128 (wireless)	1-256
Reauthentication period	3600	0 -86400
Retransmission	enable	disable
Retransmission count	3	1-16
Retransmission period	3	1-120
Silent period	5	0-3600

Configuring 802.1X Port Control

You can configure how an Ethernet interface deals with 802.1X authentication attempts. By default, the port state is **auto**, which allows 802.1X authentication to proceed normally. You can configure the Ethernet interface so that it blocks all traffic and ignores all attempts by clients to authenticate by using the **force-unauthorized** option. You can also configure the interface to successfully authenticate all attempts by clients (also known as force-authorized state) by disabling 802.1X for the interface.

In the following examples, you set the port-control state to force-unauthorized for the ethernet1 interface, which specifies that the interface blocks all traffic and ignores client authentication attempts.

WebUI

Network > Interfaces > List > Edit (for Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Port Control: Select **Force-unauthorized**.

CLI

```
set interface ethernet1 dot1x port-control force-unauthorized
```

Configuring 802.1X Control Mode

You can specify whether MAC address-based authentication is performed on devices connected to the interface by specifying one of the following modes:

- **Interface:** MAC addresses of devices connected to the interface are not authenticated. Use this option only if one trusted device is connected to the interface.
- **Virtual:** MAC addresses of devices connected to the interface are authenticated. Packets from devices with unauthorized MAC addresses are dropped. This mode is the default for an interface. Wireless interfaces use only virtual mode.

In the following examples, you set the control mode to interface for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Control Mode: Interface

CLI

```
set interface ethernet1 dot1x control-mode interface
```

Setting the Maximum Number of Simultaneous Users

When an interface is in virtual control mode, the security device allows up to the configured number of simultaneous users. By default, the security device accepts 16 simultaneous users for Ethernet interfaces or 128 users for wireless interfaces. The valid value range is 1 through 256. If you have configured the control mode to interface mode, you cannot configure the maximum number of simultaneous users.

In the following examples, you set the maximum number of simultaneous users for the ethernet1 interface to 24.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Maximum User: 24

CLI

```
set interface ethernet1 dot1x max-user 24
```

Configuring the Reauthentication Period

By default, reauthentication of 802.1X supplicants (clients) is enabled on the security device. The security device attempts to reauthenticate clients after 3600 seconds (1 hour).

For Ethernet interfaces, you can configure the reauthentication period from 0 through 86400 seconds (24 hours). To disable the reauthentication period, set the period to 0. For wireless interfaces, you cannot change the reauthentication period from its default value. If a RADIUS server provides a reauthentication period other than the default value, the security device can use the RADIUS-assigned value.

In the following examples, you set the reauthentication period to 7200 seconds (2 hours) for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Authentication Period: 7200

CLI

```
set interface ethernet1 dot1x reauth-period 7200
```

To set the reauthentication period to its default value, use the **unset interface *interface_name* dot1x reauth-period** command.

Enabling EAP Retransmissions

You can enable the retransmission of EAP requests to a client if it does not respond. By default, retransmission is enabled. Optionally, you can also configure the maximum number of EAP requests that are retransmitted and the time that elapses between retransmissions. If the maximum number of retransmissions is reached, the client's authenticated session is terminated, and authentication fails.

In the following examples, you enable the retransmission of EAP requests for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Select the 8021.X Enable check box, then click **Apply**.

CLI

```
set interface ethernet1 dot1x retry
```

Configuring EAP Retransmission Count

By default, the security device sends up to three EAP requests. You can configure the number of EAP requests from 1 through 16.

In the following examples, you set the number of EAP request transmissions to 8.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Transmission Count: 8

CLI

To configure the number of retransmit packets sent, use the following command:

```
set interface ethernet1 dot1x retry count 8
```

Configuring EAP Retransmission Period

By default, period between EAP retransmissions is 3 seconds. You can configure a period from 1 through 120 seconds.

In the following examples, you set the period between EAP retransmissions to 5.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Transmission Period: 5

CLI

```
set interface ethernet1 dot1x retry period 5
```

Configuring the Silent (Quiet) Period

The silent (quiet) period is the amount of time the security device remains silent after authentication has failed. During the silent period, the security device does not initiate or respond to any client authentication requests.

By default, when authentication fails, the security device is silent for 5 seconds, and the authentication retry count resets to zero (0).

The silent period is a value from 0 through 3600 seconds (1 hour). The 802.1X authentication state remains unauthorized after the retry fails if you specify a silent period of zero (0).

In the following examples, you set the silent period to 30 seconds for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Silent Period: 30

CLI

```
set interface ethernet1 dot1x silent-period 30
```

Configuring Authentication Server Options

If you have configured authentication servers in your network and defined them, you can specify one of these servers as the authentication server for an interface. You can also set the following authentication server options:

- Account type (802.1X clients)
- Zone verification

Specifying an Authentication Server

You can use a predefined server as the authentication server for a specific interface.

Ethernet Interfaces

For Ethernet interfaces, you specify an authentication server by modifying the interface configuration. In the following examples, you specify the existing **radius1** server as the authentication server for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Select the authentication server from the Server Name list, then click **Apply**.

CLI

```
set interface ethernet1 dot1x auth-server radius1
```

Wireless Interfaces

For wireless interfaces, you specify an authentication server by modifying the SSID configuration. In the following examples, you modify the SSID named **hr** and specify the existing **radius1** server as the authentication server for the wireless0/1 interface.

Wireless > SSID > Edit (for hr SSID): Click **New**. Depending on the security device, select one of the following, then click **OK**.

WEP Based Authentication and Encryption Methods: WEP Encryption: Select **Open**; **WEP Encryption**; and **radius1** from the Auth Server list.

802.1X Based Authentication and Encryption Methods: Select **802.1X** and then select **radius1** from the Auth Server list.

CLI

```
set ssid hr authentication wpa encryption auto auth-server radius1
```

To use the WebUI to configure authentication server information, navigate to the Auth Servers page:

Configuration > Auth > Auth Servers: Enter or select the applicable option value, then click **Apply**.

Setting the Account Type

When defining an authentication server or modifying it, you can specify the authentication server to accept 802.1X clients.

WebUI

To specify that the authentication server accept 802.1X clients, use the following procedure:

Configuration > Auth > Auth Servers > New (or Edit for existing server): Enter all relevant information; in the Account Type area, select the 802.1X check box; click **Apply**.

CLI

To specify that the existing server named **radius1** accept 802.1X clients, enter the following command:

```
set auth-server radius1 account_type 802.1x
```

Enabling Zone Verification

If your RADIUS server supports vendor-specific attribute (VSA) enhancement, you can enable zone verification, which verifies the zones the user is a member of and the zone configured on the port. Authentication is allowed only if the zone configured on the port is a zone that a user is a member of.

In your dictionary file, add an attribute name of Zone_Verification as a string attribute type. The vendor ID is 3224, and the attribute number is 10.

WebUI

To enable zone verification, use the following procedure:

Configuration > Auth > Auth Servers > New (or Edit for existing server): Enter all relevant information; select RADIUS; select the Enabled check box for Zone Verification; click **Apply**.

CLI

To enable zone verification for the RADIUS server named **radius1**, enter the following command:

```
set auth-server radius1 radius zone-verification
```

Viewing 802.1X Information

You can view detailed information about 802.1X configuration in the CLI. Not all 802.1X information can be viewed using the WebUI.

Viewing 802.1X Global Configuration Information

Enter the following command to view the global 802.1X configuration information:

```
get dot1x
```

The command shows the following information for each Ethernet and wireless interface:

- 802.1X status: enabled or disabled
- Mode: virtual or interface
- Number of users out of the maximum number of users allowed
- Number of seconds until reauthentication is required
- Port-control mode status

The following is sample output for the **get dot1x** command:

```
-----
```

Name	IEEE802.1x	Mode	User	re-auth	Status
Ethernet1	Enabled	virtual	1/64	3600s	Auto
Ethernet2	Disabled	virtual	0/64	3600s	Auto
Ethernet3	Enabled	interface	--	1200s	F-U
Ethernet3.1	Enabled	virtual	0/64	3600s	Auto
Ethernet4	Enabled	virtual	0/16	3600s	Auto

```
-----
```

Viewing 802.1X Information for an Interface

Enter the following command to view 802.1X configuration and user information for a specific interface:

```
get interface interface_name dot1x
```

The following is sample output:

```
IEEE 802.1x enabled
port-control: auto, mode: virtual
user 1/max 64 auth-server: test-radius
reauth enable period 1200s
silent enable period 300s
to-supPLICANT retry enable count 3 period 10s
-----
User 0003e40220b1, session id 1, authorized
Total 1 user shown
-----
```

Viewing 802.1X Statistics

Use the WebUI or CLI to get 802.1X statistics for a specific interface.

WebUI

Network > 802.1X > Statistic: Select the interface from the list at the top of the page.

CLI

Enter the following command to view 802.1X statistics for the ethernet0/2 interface:

```
get interface ethernet0/2 dot1x statistics
```

The following is sample output:

```
Interface Ethernet0/2:
-----
Interface ethernet1 802.1x statistics:
in eapol          0 | out eapol          0 | in start          0
in logoff         0 | in resp/id        0 | in resp           0
out req/id        0 | out req           0 | in invalid        0
in len error      0 |
Interface ethernet1 802.1x diagnostics:
while connecting:
enters            0 | eap logoffs       0 |
while authenticating:
enters            0 | auth success      0 | auth timeouts     0
auth fail         0 | auth reauth       0 | auth start        0
auth logoff       0 |
```

Viewing 802.1X Session Statistics

Enter the following command to view 802.1X session statistics:

```
get dot1x session
```

The following is sample output:

```
Alloc 2/max 1024, alloc failed 0
Id 1/ vsys 0, flag 00000000, re-auth 3105s, ethernet1, 0003e40220c2, authorized
Id 2/ vsys 0, flag 00000000, re-auth 430s, ethernet3.1, 0003e40220b1,
fail-silent
Total 2 session shown
```

Viewing 802.1X Session Details

Enter the following command to view detailed information for a specific 802.1X session:

```
get dot1x session id session_id
```

The following is sample output:

```
Id 1, flag 00000000, vsys id 0(Root)
Interface ethernet1(vsd 0), supp-mac 0003e40220c2, status authorized
Re-auth timeout 3105s, type eap-md5
  As radius_test, zone-verification on
  Retry 0, as retry 0
-----
statistics:
in octets          0 | out octets          0 | in frames          0
out frames         0
```

Configuration Examples

This section contains the following three examples:

- “Configuring the Security Device with a Directly Connected Client and RADIUS Server” on page 102
- “Configuring a Security Device with a Hub Between a Client and the Security Device” on page 103
- “Configuring the Authentication Server with a Wireless Interface” on page 104

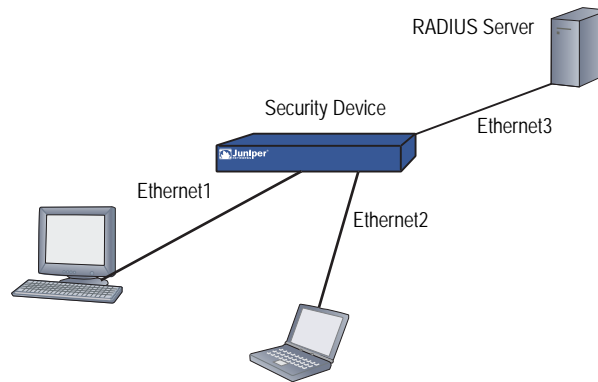
Configuring the Security Device with a Directly Connected Client and RADIUS Server

This network scenario, as shown in Figure 25, has two clients directly connected to the security device with the following parameters:

- Client directly connected to Ethernet1 interface
- Client directly connected to Ethernet2 interface
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret

Because the two clients directly connected are the only devices connected to the Ethernet1 and Ethernet2 interfaces, the control-mode is configured to **interface**.

Figure 25: Security Device with a Directly Connected Client and RADIUS Server



```

set interface ethernet1 dot1x
set interface ethernet2 dot1x
set interface ethernet1 dot1x control-mode interface
set interface ethernet2 dot1x control-mode interface

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24

set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

set interface ethernet1 dot1x auth-server radius1
set interface ethernet2 dot1x auth-server radius1

```

Configuring a Security Device with a Hub Between a Client and the Security Device

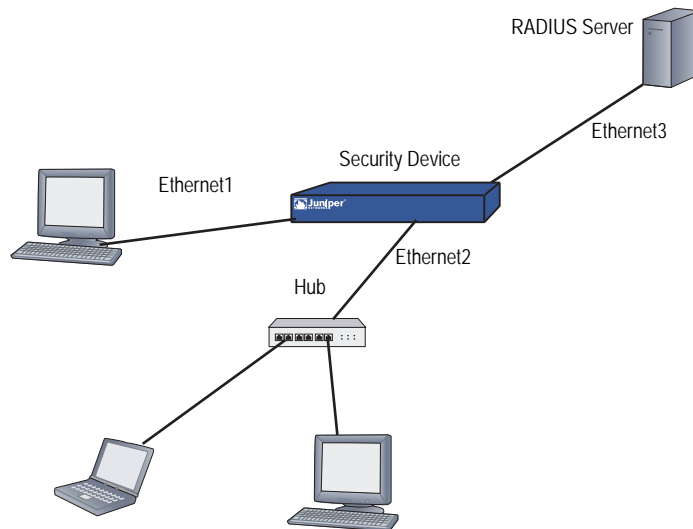
The following scenario, as shown in Figure 26, uses a hub with attached clients connected to the security device and a client directly connected to the security device.

NOTE: 802.1X functionality is not supported for a switch between the security device and clients. If you have a switch connected to the security device, we recommend disabling 802.1X on the interface to which the switch is connected.

This scenario uses the following parameters:

- Hub connected to Ethernet2 interface (control-mode of **virtual**)
- Client directly connected to Ethernet1 interface (control-mode of **interface**)
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret

Figure 26: Security Device with a Hub Between a Client and the Security Device



```

set interface ethernet1 dot1x
set interface ethernet2 dot1x
set interface ethernet1 dot1x control-mode interface
set interface ethernet2 dot1x control-mode virtual

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24

set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

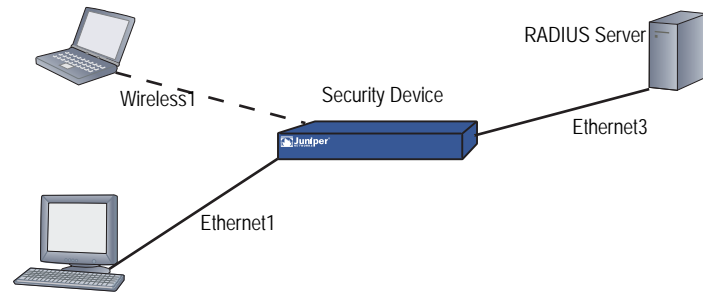
set interface ethernet1 dot1x auth-server radius1
set interface ethernet2 dot1x auth-server radius1
    
```

Configuring the Authentication Server with a Wireless Interface

The following scenario, as shown in Figure 27, has a security device with a wireless interface serving wireless clients and a client directly connected to the security device with the following parameters:

- Wireless clients connected to the wireless interface
- Client directly connected to Ethernet1 interface (control-mode of **interface**)
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret
- SSID named engineering, using WPA authentication, either AES or TKIP encryption, specifying radius1 as the authentication server, and bound to wireless interface 1

Figure 27: Configuring an Authentication Server with a Wireless Interface



```

set interface ethernet1 dot1x
set interface ethernet1 dot1x control-mode interface

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24

set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

set interface ethernet1 dot1x auth-server radius1

set ssid name engineering
set ssid engineering authentication wpa encryption auto auth-server radius1
set ssid engineering interface wireless0/1
  
```


Index

A	
addresses	
IP, lifetime for XAuth users.....	72
L2TP assignments.....	86
addresses, XAuth	
assignments.....	70
authentication, and.....	81
timeout.....	72
admin users.....	2
authentication, prioritizing.....	32
privileges from RADIUS.....	2
server support.....	14
timeout.....	18
auth servers.....	13 to 40
addresses.....	18
authentication process.....	17
backup.....	18
default.....	39, 40
defining.....	33 to 40
external.....	17
ID number.....	18
idle timeout.....	18
LDAP.....	29 to 30
maximum number.....	14
objects.....	18
SecurID.....	27
SecurID, defining.....	35
TACACS+ , defining.....	38
types.....	18
XAuth queries.....	71
auth servers, RADIUS.....	19 to 22
defining.....	33
user-type support.....	20
auth users.....	47 to 66
admin.....	2
groups.....	47, 50
IKE.....	14, 67
in policies.....	48
L2TP.....	86
local database.....	15 to 16
logins, with different.....	5
manual key.....	14
multiple-type.....	4
server support.....	14
timeout.....	18
types and applications.....	1 to 5
user types.....	13
WebAuth.....	14
XAuth.....	70
auth users, authentication	
auth servers, with.....	14
point of.....	1
pre-policy.....	49
auth users, run-time	
auth process.....	48
authentication.....	48
user groups, external.....	55
user groups, local.....	52
users, external.....	54
users, local.....	51
auth users, WebAuth.....	49
user groups, external.....	61
user groups, local.....	60
with SSL (user groups, external).....	63
authentication	
prioritizing.....	32
authentication servers	
See auth servers	
authentication users	
See auth users	
B	
banners.....	10
bypass-auth.....	71
C	
CHAP.....	81
clusters, Unified Access Control.....	41
common names.....	30
connection policy for Infranet Enforcer, configuring.....	42
D	
databases, local.....	15 to 16
dictionary file, RADIUS.....	2
distinguished names.....	30
E	
encryption	
SecurID.....	28

F		P	
fallback priorities, assigning.....	32	priorities, assigning	32
		protocols, CHAP	81
G		R	
group expressions	5 to 9	RADIUS	19 to 22
operators.....	5	auth server objects.....	33
server support	14	dictionary file.....	2
users	5	dictionary files	21
		object properties	20
I		ports.....	20
idle session timeout	18	retry timeout.....	20
IKE users.....	14, 67 to 70	shared secret	20
defining	68	Remote Authentication Dial-in User Service	
groups.....	68	<i>See</i> RADIUS	
groups, and.....	67	RFC 1777, <i>Lightweight Directory Access Protocol</i>	29
groups, defining	69	run-time authentication	48
IKE ID	67, 81		
server support	14	S	
with other user types.....	4	ScreenOS	
Infranet Controller		RADIUS vendor IDs.....	22
actions	43	SecurID	27
overview.....	42	ACE servers.....	28
resource policies	43	auth server object.....	35
Infranet Enforcer		authentication port	28
connection policy, configuring.....	42	authenticator.....	27
overview.....	42	encryption types	28
		token codes.....	27
L		Use Duress option.....	28
L2TP		user type support	28
address assignments	86	SecurID clients	
external auth server.....	86	retries.....	28
local database	86	timeout	28
user authentication	86	servers, auth	
L2TP users	86	<i>See</i> auth servers	
server support	14	servers, SecurID ACE.....	28
with XAuth.....	4	session idle timeout	18
LDAP	29 to 30	SSL, with WebAuth	64
common name identifiers	30		
distinguished names.....	30	T	
server ports.....	30	TACACS+	
structure	29	auth server objects.....	38
user types supported	30	clients retries	32
Lightweight Directory Access Protocol		clients timeout.....	32
<i>See</i> LDAP		object properties	32
local database		ports.....	32
IKE users	68	retry timeout.....	32
timeout.....	16	shared secret	32
user types supported	16	timeout	
		admin users	18
M		auth users.....	18
manual keys.....	14	token codes	27
mode config.....	71	traffic	
		redirecting HTTP with WebAuth	50

- U**
- UAC clusters..... 41
 - Unified Access Control (UAC) 41
 - unified access control solution
 - overview of vii
 - users
 - admin 2
 - admin, timeout..... 18
 - groups, server support..... 14
 - IKE
 - See IKE users
 - L2TP 86 to 89
 - multiple-type..... 4
 - WebAuth 14
 - XAuth..... 70 to 84
 - users, auth
 - See auth users
 - users, IKE
 - See IKE users
- V**
- vendor IDs, VSA 22
 - vendor-specific attributes 21
 - virtual adapters 70
 - VPN idletime 73
 - VSA attribute types 22
 - VSAs 21
- W**
- WebAuth 14, 49
 - external user groups 61
 - pre-policy auth process 49
 - redirecting HTTP traffic 50
 - user groups, local 60
 - with SSL (user groups, external)..... 63
- X**
- XAuth
 - bypass-auth..... 71
 - client authentication 85
 - defined 70
 - query remote settings..... 71
 - ScreenOS as client 85
 - TCP/IP assignments 72
 - virtual adapters..... 70
 - VPN idletime..... 73
 - XAuth addresses
 - assignments..... 70
 - authentication, and..... 81
 - IP address lifetime 72 to 73
 - timeout 72
 - XAuth users 70 to 84
 - authentication..... 70
 - local authentication..... 73
 - local group authentication..... 75
 - server support..... 14
 - with L2TP 4
 - XAuth, external
 - auth server queries..... 71
 - user authentication 76
 - user group authentication 78

